



Simulador de Sistema de Informação ao Público para Rede de Transportes Metro- Ferroviária

NUNO MIGUEL NUNES DA ROCHA
Novembro de 2013

SIMULADOR DE SISTEMA DE INFORMAÇÃO AO PÚBLICO PARA REDE DE TRANSPORTES METRO-FERROVIÁRIA

Nuno Miguel Nunes da Rocha



Mestrado em Engenharia Eletrotécnica e de Computadores

Área de Especialização de Telecomunicações

Departamento de Engenharia Eletrotécnica

Instituto Superior de Engenharia do Porto

2013

Este relatório satisfaz, parcialmente, os requisitos que constam da Ficha de Disciplina de Tese/Dissertação, do 2º ano, do Mestrado em Engenharia Eletrotécnica e de Computadores

Candidato: Nuno Miguel Nunes da Rocha, N° 1020877, 1020877@isep.ipp.pt

Orientação científica: Dr.ª Paula Maria Marques Moura Gomes Viana, pmv@isep.ipp.pt

Empresa: EFACEC Engenharia e Sistemas, S.A.

Supervisão: Eng.º Joaquim Jorge Ferreira Nunes, joaquim.nunes@efacec.com



Mestrado em Engenharia Eletrotécnica e de Computadores

Área de Especialização de Telecomunicações

Departamento de Engenharia Eletrotécnica

Instituto Superior de Engenharia do Porto

19 de Novembro de 2013

Dedico este trabalho à minha família pois sem o seu apoio nada disto teria sido possível.

Agradecimentos

Agradeço à Dr.^a Paula Viana pelo apoio e disponibilidade prestados durante a realização deste trabalho. Tenho que agradecer também ao Eng.^a Joaquim Nunes por todo o apoio prestado, sem o qual não teria sido possível elaborar este projeto.

Agradeço também à empresa EFACEC por esta oportunidade, pois permitiu-me adquirir experiência em ambiente laboral e ainda aos colegas de estágio Carlos Faria, Luís Conceição, Daniel Rocha e João Peixoto pelo apoio prestado.

Resumo

O aumento da complexidade e das funcionalidades a fornecer pelos equipamentos de apoio à exploração de sistemas de redes de transporte criou um novo paradigma no qual vários intervenientes são chamados a fornecer módulos. Estes devem comunicar entre si de forma transparente, não devendo haver limitações colocadas pela tecnologia utilizada ou pelo facto de serem de fabricantes distintos.

Este projeto de dissertação resulta de uma proposta da empresa EFACEC para a construção de um Simulador de um Sistema de Informação ao Público numa rede de transportes metro-ferroviária, que utilize como interface para comunicação com a plataforma de gestão EFARail a tecnologia de *Web Services*.

Nesta dissertação faz-se um levantamento dos principais equipamentos utilizados em Sistemas de Informação ao Público e das funcionalidades disponibilizadas, assim como das tecnologias de *Web Services* que permitem responder aos requisitos e funcionalidades gerais do sistema.

A aplicação desenvolvida permite a simulação duma rede de equipamentos visuais e sonoros de informação ao público que compõem uma linha metro-ferroviária. Desta forma é possível testar novos módulos sem necessidade de uma integração em ambiente real. Com isto pretende-se otimizar o processo de desenvolvimento, diminuindo os custos associados ao teste dos sub-sistemas que compõem a solução.

Palavras-Chave

Web Services, Sistemas de Informação ao Público, SOAP, WSDL, Java, JSP, EFARail.

Abstract

The increase in complexity and the functionalities to provide by equipment to support the operation of systems of transport networks has created a new paradigm in which various actors are called to provide modules. These must communicate among themselves in a transparent way, there should be no limitations placed by technology used or by the fact that they are of distinct manufacturers.

This dissertation project results from a proposal by the company EFACEC to build a Simulator of a Public Information System transport network in metro-rail, which use as interface for communication with the management platform EFARail the technology of Web Services.

This dissertation makes a survey of the main equipment used in Public Information Systems and features offered, as well as the technologies of Web Services that allow them to respond to the requirements and general features of the system.

The developed application allows the simulation of a visual and audio equipment network of information to the public that make up a metro-rail line. This way it is possible to test new modules without need for an integration in a real environment. With this we intend to optimize the development process, reducing costs associated with test of sub-systems that make up the solution.

Keywords

Web Services, Public Information Systems, SOAP, WSDL, Java, JSP, EFARail.

Índice

AGRADECIMENTOS	I
RESUMO	III
ABSTRACT	V
ÍNDICE	VII
ÍNDICE DE FIGURAS	IX
ÍNDICE DE TABELAS	XI
ACRÓNIMOS.....	XIII
1. INTRODUÇÃO	1
1.1. CONTEXTUALIZAÇÃO	1
1.2. OBJETIVOS.....	2
1.3. ORGANIZAÇÃO DO RELATÓRIO	3
2. SISTEMAS DE INFORMAÇÃO AO PÚBLICO	5
2.1. NECESSIDADES DE INFORMAÇÃO DO PASSAGEIRO	5
2.2. CONSTITUIÇÃO DOS SISTEMAS DE INFORMAÇÃO AO PÚBLICO	8
2.3. MECANISMOS DE DISPONIBILIZAÇÃO DE INFORMAÇÃO AO PÚBLICO	10
2.4. SIRI (<i>SERVICE INTERFACE FOR REAL-TIME INFORMATION</i>).....	13
2.5. INTEGRAIL – PROJETO EUROPEU PARA INTEGRAÇÃO DE TRANSPORTES FERROVIÁRIOS	14
2.6. SISTEMA DE INFORMAÇÃO AO PÚBLICO DA EFACEC	15
3. TECNOLOGIA DE WEB SERVICES PARA A CONSTRUÇÃO DE SISTEMAS DE INFORMAÇÃO AO PÚBLICO.....	19
3.1. ARQUITETURA DE UM WEB SERVICE	21
3.2. PROTOCOLOS E LINGUAGENS ASSOCIADOS AOS WEB SERVICES	23
3.3. TECNOLOGIAS MAIS UTILIZADAS NA CONSTRUÇÃO DE WEB SERVICES.....	25
4. IMPLEMENTAÇÃO	29
4.1. REQUISITOS DO SISTEMA	29
4.2. PROPOSTA DE ARQUITETURA	30
4.3. ARQUITETURA DO SOFTWARE	31
4.4. INTERFACE COM O UTILIZADOR	38
5. TESTES FUNCIONAIS.....	47
6. CONCLUSÕES.....	51
REFERÊNCIAS DOCUMENTAIS	53

ANEXO A. BASE DE DADOS DESENVOLVIDA.....	55
ANEXO B. TESTES EFETUADOS AO MÉTODO DISPLAYMESSAGE.	57

Índice de Figuras

Figura 1	Interfaces entre sistemas e SFP.	2
Figura 2	Constituição de um sistema de informação ao público.	8
Figura 3	Terminal interativo.	11
Figura 4	Painel digital.	11
Figura 5	Arquitetura global da solução EFACEC.	15
Figura 6	EFARail plataforma de operação e gestão para ambientes metro-ferroviários da EFACEC.	17
Figura 7	Arquitetura de interação com um <i>Web Service</i>	21
Figura 8	Arquitetura dos <i>Web Services</i>	22
Figura 9	Comunicação Cliente/Servidor utilizando JAX-RPC.	25
Figura 10	Arquitetura de um <i>Web Service</i> construído na plataforma .NET.	26
Figura 11	Arquitetura do simulador do SIP.	30
Figura 12	Diagrama de blocos do Simulador do SIP.	33
Figura 13	Arquitetura do software do Simulador do SFP.	38
Figura 14	Página de entrada da interface gráfica do simulador do SIP.	39
Figura 15	Informação obtida através da opção LOGs.	40
Figura 16	Informação obtida através da opção FERIADOS.	40
Figura 17	Informação obtida através da opção PLAYLISTs.	41
Figura 18	Informação obtida através da opção MENSAGENS PRÉ-GRAVADAS.	41
Figura 19	Informação obtida através da opção EVENTOS TEMPORÁRIOS.	42
Figura 20	Informação obtida através da opção GRUPOS DE EQUIPAMENTOS.	42
Figura 21	Informação obtida quando se seleciona um equipamento visual.	43
Figura 22	Página de entrada da interface gráfica do Simulador do SFP.	43
Figura 23	Formulário disponibilizado pela opção Afixar Mensagem.	44
Figura 24	Página disponibilizada quando se seleciona a opção Equipamento Sonoro.	45
Figura 25	Página disponibilizada quando se seleciona a opção Estado dos Equipamentos.	45
Figura 26	Criação de um novo projeto.	47
Figura 27	Árvore com os métodos disponíveis pelo <i>Web Service</i>	48
Figura 28	Seleção do método GetHolidays.	48
Figura 29	Mensagens SOAP trocadas.	49
Figura 30	Base de Dados desenvolvida.	55
Figura 31	Comando enviado através do Simulador do SFP.	57
Figura 32	Comando enviado através do software soapUI.	58

Índice de Tabelas

Tabela 1	Descrição das tecnologias utilizadas em sistemas distribuídos.....	20
Tabela 2	Resumo das normas e protocolos associados aos <i>Web Services</i>	23
Tabela 3	Descrição das principais tabelas da base de dados implementada.	31
Tabela 4	Descrição dos ficheiros contidos no diretório Equipamento Sonoro	34
Tabela 5	Descrição dos ficheiros contidos no diretório Equipamento Visual.....	34
Tabela 6	Descrição dos ficheiros contidos no diretório SIMSIP	35
Tabela 7	Classes que constituem o bloco Lógica da Aplicação.....	35
Tabela 8	Classes que constituem o bloco Acesso à Base de Dados.....	37

Acrónimos

API	– <i>Application Programming Interface</i>
COM	– <i>Component Object Model</i>
CORBA	– <i>Common Object Request Broker Architecture</i>
DCOM	– <i>Distributed Component Object Model</i>
FTP	– <i>File Transfer Protocol</i>
HTTP	– <i>Hypertext Transfer Protocol</i>
IDE	– <i>Integrated Development Environment</i>
JAXB	– <i>Java Architecture for XML Binding</i>
JAX-RPC	– <i>Java API for XML-based RPC</i>
JAX-WS	– <i>Java API for XML Web Services</i>
JSP	– <i>Java Server Pages</i>
RMI	– <i>Remote Method Invocation</i>
RPC	– <i>Remote Procedure Call</i>
SCAP	– <i>Sistema de Controlo de Acesso e de Passageiros</i>
SCT	– <i>Sistema de Controlo de Trens</i>
SFP	– <i>Sistema de Fluxo de Passageiros</i>
SIP	– <i>Sistema de Informação ao Público</i>
SIRI	– <i>Service Interface for Real-time Information</i>
SMTP	– <i>Simple Mail Transfer Protocol</i>
SOA	– <i>Service Oriented Architecture</i>
SOAP	– <i>Simple Object Access Protocol</i>
UDDI	– <i>Universal Description, Discovery and Integration</i>
URI	– <i>Uniform Resource Identifier</i>
XML	– <i>eXtensible Markup Language</i>
WSDL	– <i>Web Service Description Language</i>

1. INTRODUÇÃO

1.1. CONTEXTUALIZAÇÃO

Num mercado cada vez mais competitivo e exigente, a capacidade de adaptação, a novas realidades e exigências, dos produtos vendidos por uma empresa é de extrema importância para o sucesso da mesma.

Na atualidade é vulgar a divisão de um projeto em vários subprojectos, sendo estes muitas das vezes elaborados por entidades distintas. Este paradigma cria uma realidade que é preciso ter em conta e que passa pela necessidade de comunicação e interoperabilidade entre os vários subsistemas que constituem o projeto, mas em particular com a plataforma centralizada de gestão e operação dos mesmos.

Para garantir que esse requisito é atingido, torna-se imprescindível a criação de mecanismos de teste e validação das interfaces de comunicação com os vários subsistemas.

Uma tecnologia que se apresenta como alternativa eficaz para a interligação entre os vários subsistemas são os *Web Services*. Esta tecnologia, através de uma comunicação baseada em protocolos e normas abertas, permite uma interoperabilidade entre os vários módulos e plataformas constituintes de um projeto.

Uma das formas de validar as interfaces e os dados transacionados entre os vários subsistemas, consiste no desenvolvimento de *software* que permita recriar cada um desses subsistemas.

1.2. OBJETIVOS

Esta dissertação está inserida no contexto dos serviços prestados pela empresa EFACEC (Unidade de Negócio de Transportes) na idealização, conceção e instalação de soluções de sistemas orientados à gestão de serviços de transportes públicos.

O objetivo específico deste trabalho é o desenvolvimento de uma aplicação que permita validar a interface e os dados transacionados entre o Sistema de Controlo de Fluxo de Passageiros (SFP) e um Sistema de Informação ao Público (SIP), seja ele próprio ou de terceiros.

O SFP consiste numa aplicação que faz parte de uma plataforma integrada de supervisão e controlo dos vários sistemas necessários à exploração de uma linha metro-ferroviária, denominada de EFARail (Figura 1).

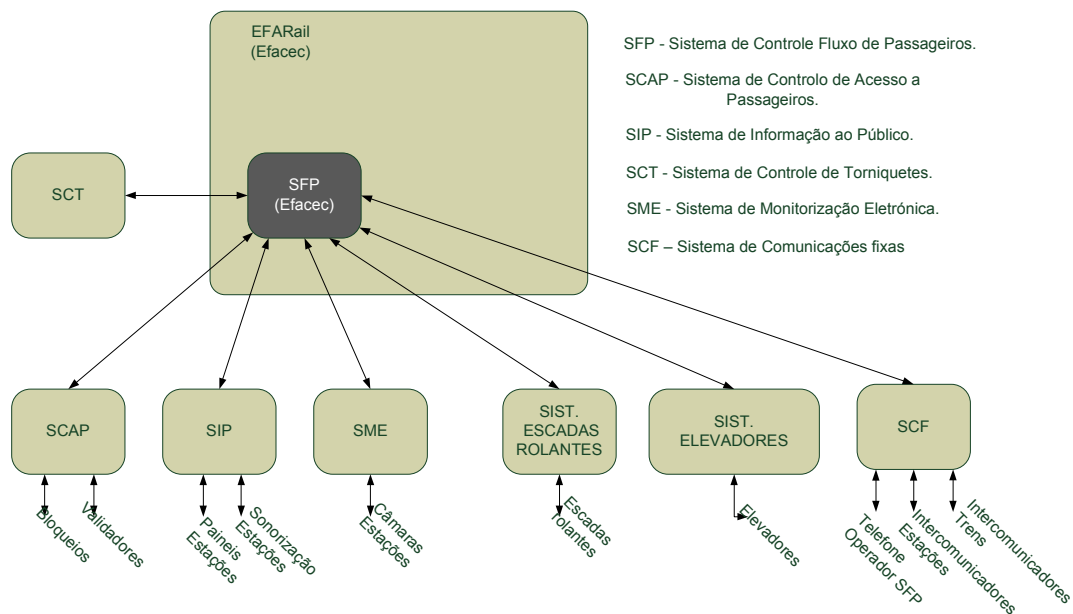


Figura 1 Interfaces entre sistemas e SFP.

Este SFP tem por objetivo o controlo de vários subsistemas constituintes de uma linha metro-ferroviária, entre os quais:

- Sistema de Controlo de Trens (SCT).
- Sistema de Informação ao Público (SIP).
- Sistema de Controlo de Acesso a Passageiros (SCAP).

- Sistema de Monitorização Eletrónica (SME) / CCTV.

O sistema a desenvolver terá como funcionalidade base a validação da interface de comunicação entre o SFP e um SIP e, ainda, a simulação dos equipamentos de informação ao público das várias estações/locais de uma linha de transporte metro-ferroviária. Serão simulados os vários equipamentos de cada uma das estações/locais, sendo apresentadas informações relativas ao estado de funcionamento dos vários equipamentos, mensagens que estão a ser afixadas ou difundidas e registo das operações efetuadas pelo SFP sobre o SIP.

Entre as principais vantagens da conceção do simulador, destaca-se a possibilidade de se efetuar testes em laboratório sem que exista a necessidade de uma ligação física entre a plataforma de gestão e o SIP. Permite ainda a validação da interface de comunicação ou a definição de uma interface própria que depois é “imposta” à entidade responsável pela instalação do SIP.

Adicionalmente, para validação do serviço, tem-se como objetivo deste trabalho a criação de um simulador do SFP que será o cliente do *Web Service* disponibilizado pelo simulador do SIP.

1.3. ORGANIZAÇÃO DO RELATÓRIO

Este relatório encontra-se estruturado em seis capítulos.

No capítulo 1 faz-se uma introdução com os objetivos estabelecidos para o desenvolvimento desta dissertação e a motivação que levou à sua realização. No capítulo 2 são apresentados alguns conceitos relacionados com os Sistemas de Informação ao Público, sendo feita uma descrição da sua constituição e das necessidades dos passageiros inerentes à sua utilização. No capítulo 3 faz-se uma descrição das principais tecnologias de *Web Services* utilizadas para a construção de Sistemas de Informação ao Público. No capítulo 4 faz-se uma descrição da arquitetura do sistema desenvolvido, descrevendo a arquitetura de *software* e as interfaces gráficas desenvolvidas. No capítulo 5 faz-se uma descrição dos vários passos necessários para a validação do *software* desenvolvido. Por último, o capítulo 6 faz-se uma análise ao trabalho desenvolvido, descrevendo as principais conclusões obtidas e perspetivando-se alguns desenvolvimentos futuros.

2. SISTEMAS DE INFORMAÇÃO AO PÚBLICO

Num sistema de transportes (metro-ferroviário ou rodoviário), a disponibilização de um sistema de informação de qualidade é de grande utilidade não só para o passageiro, mas também para o operador de transportes. Tal sistema permite consolidar a relação com os clientes, potencia o acesso a novos clientes e facilita o ajustamento entre a oferta e a procura [1].

2.1. NECESSIDADES DE INFORMAÇÃO DO PASSAGEIRO

Um SIP é naturalmente uma parte essencial e inerente do transporte público. Não se pode esperar que as pessoas viajem num transporte público, se elas não souberem para onde podem viajar, que tipo de serviços é que estão disponíveis e onde é que podem aguardar pela chegada desse transporte. Igualmente importante, as pessoas precisam de saber o custo da viagem.

Estes elementos são de extrema importância e a precisão das informações nesta fase pode ter um impacto significativo sobre a perceção da viagem que o passageiro pretende fazer. Informações erradas são tão prejudiciais como não ter qualquer tipo de informação, na ver-

dade há argumentos plenamente justificados que defendem que fornecer informações erradas é pior do que não fornecer qualquer tipo de informação.

O passageiro experiente, cada vez mais exige melhor informação e de fácil acesso. O advento das comunicações móveis permitiu que fosse fornecida informação personalizada quer antes do início da viagem como durante a mesma [2].

Um sistema de informação de qualidade para o passageiro é o que lhe permite:

- Obter uma rápida percepção da oferta disponível para as suas necessidades.
- Planear a sua viagem conhecendo as alternativas do sistema (percursos, horários e preço).
- Saber com fiabilidade as horas de chegada do seu transporte.
- Ser informado sobre acontecimentos inesperados (interrupções/alterações do serviço).
- Ser informado sobre novos serviços/novas oportunidades para satisfazer as suas necessidades de mobilidade.
- Dispor de suportes informativos cada vez mais acessíveis e eficazes.

Um sistema de informação ao passageiro segue uma sequência de três etapas, que são:

- Informação adquirida antes da viagem.
- Informação adquirida durante a viagem.
- Informação adquirida depois da viagem.

Para cada um destes contextos o passageiro tem várias finalidades para cumprir.

2.1.1. INFORMAÇÃO ADQUIRIDA ANTES DA VIAGEM

A informação recolhida pelo passageiro antes do início de uma viagem permite-lhe identificar o melhor trajeto, os custos associados à viagem e o modo de reserva de bilhetes, os horários de chegada e partida de veículos, as ligações entre redes de transporte e as ligações entre veículos da mesma rede.

A escolha do transporte público está diretamente ligada ao tipo e qualidade de informação existente. Logo, se a informação existente para um determinado destino for pouca, poderá

levar a que o passageiro opte pelo meio de transporte privado ou até a cancelar a viagem. Os fatores que têm influência na escolha do meio de transporte e da rota são muitos e dependem do passageiro, da sua experiência e da finalidade da viagem [2].

O passageiro compara diferentes alternativas (se houver mais do que uma) por preço, horário e informações de conexão, clareza da rota, o conforto e os serviços a bordo.

Assim, a via escolhida pode ser a mais curta, mais rápida, mais barata, mais familiar ou aquela com o menor número de transferências.

2.1.2. INFORMAÇÃO FORNECIDA DURANTE A VIAGEM

Geralmente as paragens, estações ou aeroportos são o primeiro contacto com a rede de transportes públicos. Estes também são os lugares onde a viagem começa e onde o passageiro realiza uma orientação e acompanhamento de tarefas, como verificação de horários, preço da viagem, etc.

Durante a viagem a informação disponibilizada permite ao passageiro saber a sua localização geográfica, através da indicação das próximas paragens, o tempo e ainda interrupções (paragem súbita da viagem, mudança de rota), atraso e seus efeitos sobre a viagem [2].

2.1.3. INFORMAÇÃO FORNECIDA DEPOIS DA VIAGEM

A informação necessária depois da viagem diz respeito à ligação entre a rede e o destino final. As necessidades são essencialmente geográficas, isto é, indicações dentro das cidades como sinais que informam a localização de atrações turísticas ou locais de interesse público [2]. Para além disso, por vezes são também necessárias informações sobre a viagem de regresso como por exemplo horários de funcionamento dos serviços de transporte público.

Um dos principais requisitos dos passageiros no final de uma viagem, recai sobre a necessidade de sair do local de paragem e atingir o local de destino pretendido. Para isso torna-se fundamental o fornecimento de indicações (sinais, mapas, mensagens sonoras entre outros) específicas que possam orientar os passageiros nesse sentido.

2.2. CONSTITUIÇÃO DOS SISTEMAS DE INFORMAÇÃO AO PÚBLICO

Um bom SIP é aquele que, utilizando vários meios de comunicação, fornece aos passageiros o acesso rápido e fácil à informação em todos os lugares onde esta possa ser necessária quando se viaja [3].

Os SIP podem ser analisados de três perspectivas:

- Uma perspectiva pragmática: porque e para que fins, são necessários os SIP?
- Uma perspectiva semântica: quais são os conteúdos dos SIP?
- Uma perspectiva sintática: como são construídos os SIP?

As três perspectivas correspondem a três plataformas como ilustrado na Figura 2 [4].

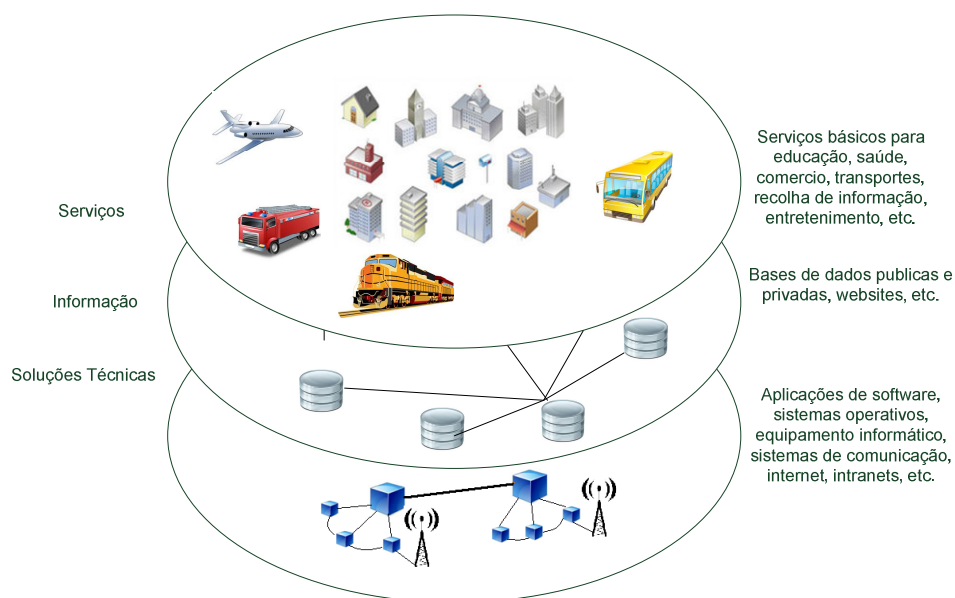


Figura 2 Constituição de um sistema de informação ao público.

2.2.1. SERVIÇOS

Os SIP não são um conceito recente, tendo sempre existido desde que os seres humanos coexistem como criaturas sociais, tentando atingir objetivos comuns de forma organizada. Com o advento dos computadores, os SIP, como acontece com todos os tipos de sistemas de informação, beneficiaram de uma nova ferramenta técnica que pode aumentar a capacidade de processamento de dados e uma disponibilização dos mesmos ao público em geral [4].

A quantidade e qualidade de informação indispensável ao funcionamento de serviços públicos de diversas áreas, traduz-se na necessidade de implementação de sistemas adaptados, que visam facilitar e conceder auxílio à realização de tarefas, conferindo maior acessibilidade, fiabilidade e segurança nos serviços prestados.

Os serviços prestados pelos SIP passam pela disponibilização de informações relacionadas com tráfego, condições meteorológicas, venda de bilhetes de transportes públicos online, entre outras.

2.2.2. INFORMAÇÃO

Os dados produzidos e utilizados por um SIP podem ser classificados de diferentes maneiras [4]:

- **Informação de operação** - São os dados que são necessários em sentido absoluto para a realização e o processamento de uma determinada tarefa, por exemplo: dados processados em *websites* durante o *login* e registo de utilizadores.
- **Informação analítica** - Informação produzida através de estatística e análise da eficiência e qualidade do processo/serviço disponibilizado.
- **Meta dados** - Informação sobre informação, isto é, caracteriza a informação que é disponibilizada ao público e auxilia na tarefa de interpretação desses dados. Estes devem auxiliar as pessoas em tarefas como recuperação de dados, interpretação de dados, execução de tarefas e operações específicas.
- **Informação de processo** - Informa se determinada transação foi processada com sucesso ou não. Pode ainda ser utilizada para sinalizar os problemas na conceção ou utilização de um SIP, e pode ser utilizada como um ponto de partida para melhorias.
- **Informação de arquivo** - São todos os registos recolhidos e armazenados que constituem um histórico de operação e interação entre os processos.

Em resumo, um SIP deve conter os dados necessários para o cumprimento das tarefas do sistema em questão.

2.2.3. SOLUÇÕES TÉCNICAS

As soluções técnicas necessárias para a construção de um sistema de informação ao público consistem em: equipamento informático (computadores, dispositivos de rede); sistema operativo que permita o acesso a *software* necessário ao processamento da informação; sistemas de comunicação (Internet, Intranets) responsáveis pela interoperabilidade entre dispositivos de hardware que constituem o sistema.

2.3. MECANISMOS DE DISPONIBILIZAÇÃO DE INFORMAÇÃO AO PÚBLICO

Cada vez mais as pessoas exigem que o serviço prestado pelos transportes públicos seja melhor. Eles querem informações precisas no momento certo, numa variedade de formatos e disponíveis para uma variedade de dispositivos. Disponibilizar uma melhor oferta nos serviços prestados pelos transportes públicos num mundo em mudança é um desafio, mas também uma oportunidade para a inovação [5].

A seguir são apresentados alguns dispositivos utilizados em sistemas de informação ao público, que auxiliam as pessoas durante uma viagem.

2.3.1. TERMINAIS PÚBLICOS INTERATIVOS

São sistemas que principalmente, fornecem informações aos passageiros antes do início da sua viagem, permitindo-lhes que tomem decisões fundamentadas sobre os trajetos dos veículos e horários de partida [6].



Figura 3 Terminal interativo.

A informação fornecida pelos terminais públicos interativos é essencialmente informação estática. Existem no entanto terminais que fornecem informação em tempo real como tráfego, estacionamento e congestionamentos nos meios de transporte. Estes também disponibilizam informação sobre a rede de transportes, as rotas, os horários e as tarifas praticadas, para que o passageiro possa planejar a sua viagem.

2.3.2. PAINÉIS DIGITAIS

Os painéis digitais exibem normalmente informações sobre a hora atual, a hora de chegada ou de partida (tempo de espera), qualquer comentário sobre a rota/linha e outras informações atualizadas que podem incluir alterações, conexões temporárias, mudanças planeadas na tabela de circulação [7].



Figura 4 Paineis digitais.

Os painéis podem ainda incluir linhas adicionais dedicadas à publicidade. Estes dispositivos devido ao fácil controlo (remoto) são ideais para a exibição de informação em tempo real, sobre alterações na circulação.

2.3.3. WEBSITES

Devido ao crescente desenvolvimento das tecnologias Web, este meio de comunicação permite uma disponibilização de conteúdos ou serviços variados. Estes conteúdos vão desde informação de horários de chegada e partida de veículos, passando pelas condições meteorológicas ou condições do tráfego, até à compra de bilhetes *online* [8].

O conceito de extensibilidade e facilidade de integração de diferentes serviços ganha extrema importância na construção de um SIP. Comparativamente aos terminais interativos, o *Website* utiliza bases de dados e interfaces semelhantes, embora sejam disponibilizados não só por operadores públicos mas também por operadores comerciais de transportes públicos.

Os *Websites* permitem que a informação seja consultada no conforto de casa ou do escritório, permitindo aos passageiros um melhor planeamento das suas viagens antes de sair de casa. Estes ao possibilitarem a ligação a outros *sites*, permitem que o passageiro possa consultar outras informações que lhe podem ser úteis para a sua viagem [9].

No entanto, apesar da capacidade de disponibilização de dados úteis associada aos *Websites*, o acesso à informação pode tornar-se mais complicado. Isto acontece quando os conteúdos não são corretamente organizados, ou quando são utilizadas tecnologias e ferramentas mais sofisticadas (como Java ou *Flash*) que dificultam a utilização do *site* por pessoas que não acedam à internet com regularidade.

2.3.4. DISPOSITIVOS PORTÁTEIS

Cada vez mais as pessoas estão a utilizar *smartphones* e *tablets* para comunicarem, acedem à internet e a *sites* com informação sobre transportes públicos. Consequentemente, é de esperar que no futuro estes dispositivos se tornem num dos mais importantes meios de disponibilização de informação ao público.

2.3.5. DISPOSITIVOS SONOROS

A difusão de mensagens é feita através de um conjunto de equipamentos de áudio ou sistema de som. Trata-se de um conjunto de equipamentos que recebe o áudio (tipicamente em *stream* digital), descodifica, amplifica-o e distribui o som para o público em geral. São normalmente usados para anúncios ou qualquer informação imediata que tem de ser comunicada, como por exemplo, em estações de caminho-de-ferro, paragens de autocarro, terminais de aeroportos e outros locais de interesse público.

2.4. SIRI (*SERVICE INTERFACE FOR REAL-TIME INFORMATION*)

Por toda a Europa, as interfaces de *software* aberto bem definidas têm um papel crucial na melhoria da viabilidade técnica e económica dos SIP de todos os tipos.

O objetivo do SIRI é permitir que uma entidade que pretenda adquirir diferentes sistemas de informação de vários fornecedores, possa interligar estes sistemas com um elevado nível de confiança de que estes irão trabalhar em conjunto [8]. Também é pretendido que exista uma proteção de longo prazo de investimento, de modo que exista um apoio contínuo para o sistema e ainda num processo coerente para a evolução dos sistemas, permitir que os componentes adicionais que podem ser adquiridos no futuro também sejam integrados no sistema.

O SIRI destina-se a ser usado para a troca de informações entre os servidores que contêm dados sobre os veículos de transporte público e dados dos tempos das viagens. Estes incluem os centros de controlo dos operadores de transporte e sistemas de informação que utilizam as informações do veículo em tempo real para operar o sistema e os sistemas a jusante que fornecem ao público informações sobre a viagem em monitores instalados nas paragens, dispositivos móveis, etc.

O SIRI especifica uma interface padrão para a troca de informações que permita avaliar o desempenho atual ou projetado de um sistema de transporte público em tempo-real entre os diferentes sistemas de computadores [10]. Esta norma é constituída por um conjunto cuidadosamente modularizado de serviços funcionais distintos para operar sistemas de informação de transportes públicos.

A modularidade desta norma permite uma abordagem incremental, isto é, somente o subconjunto de serviços realmente necessários precisa de ser implementado para uma deter-

minada aplicação. A expectativa é que os utilizadores comecem com apenas um ou dois serviços e ao longo do tempo aumentem o número de serviços e a gama de opções suportadas. Da mesma forma os fornecedores de serviços podem estender o seu suporte ao SIRI de forma incremental.

Todos os serviços prestados pelo SIRI são construídos sobre uma camada normalizada, baseada numa arquitetura de *Web Services* [11] e utiliza XML (*eXtensible Markup Language*) para definir as suas mensagens. É feita uma separação cuidadosa entre o transporte (como os dados são transportados) e a quantidade de dados acrescentado ao cabeçalho das mensagens. Deste modo as mensagens SIRI podem ser trocadas tanto como documentos XML utilizando o método POST do protocolo HTTP (*Hypertext Transfer Protocol*) ou utilizando o protocolo SOAP (*Simple Object Access Protocol*) [11].

2.5. INTEGRAIL – PROJETO EUROPEU PARA INTEGRAÇÃO DE TRANSPORTES FERROVIÁRIOS

O INTEGRAIL foi um projeto Europeu destinado a criar um sistema de informação coerente, integrando os principais sub-sistemas ferroviários. Este projeto teve como objetivo o aumento do desempenho do sistema ferroviário em termos de capacidade, velocidade média e pontualidade, segurança e otimização na utilização dos recursos [12].

As principais recomendações feitas pelo projeto consistem na utilização de um modelo de dados que fornecesse o suporte para a utilização de uma Arquitetura Orientada aos Serviços (SOA – *Service Oriented Architecture*).

O SOA é uma abordagem em termos de arquitetura corporativa, que permite a criação de serviços de negócio interoperáveis que podem facilmente ser reutilizados e compartilhados entre aplicações e empresas. Dentro de uma arquitetura SOA todos os processos de negócio são desenvolvidos dentro de uma estrutura, através da qual eles são capazes de comunicar com outros processos para proporcionar uma operação de negócios abrangente.

Normalmente uma arquitetura SOA é construída utilizando *Web Services* e isso é verificado nas propostas apresentadas para o projeto INTEGRAIL [12].

2.6. SISTEMA DE INFORMAÇÃO AO PÚBLICO DA EFACEC

Os SIP desenvolvidos pela EFACEC assentam numa solução cuja arquitetura global se baseia num centro de controlo que concentra a operação e gestão do sistema e interage com os equipamentos periféricos, geograficamente dispersos pelas estações. Esta solução permite disponibilizar aos utentes informações nos formatos visual e sonoro [13].

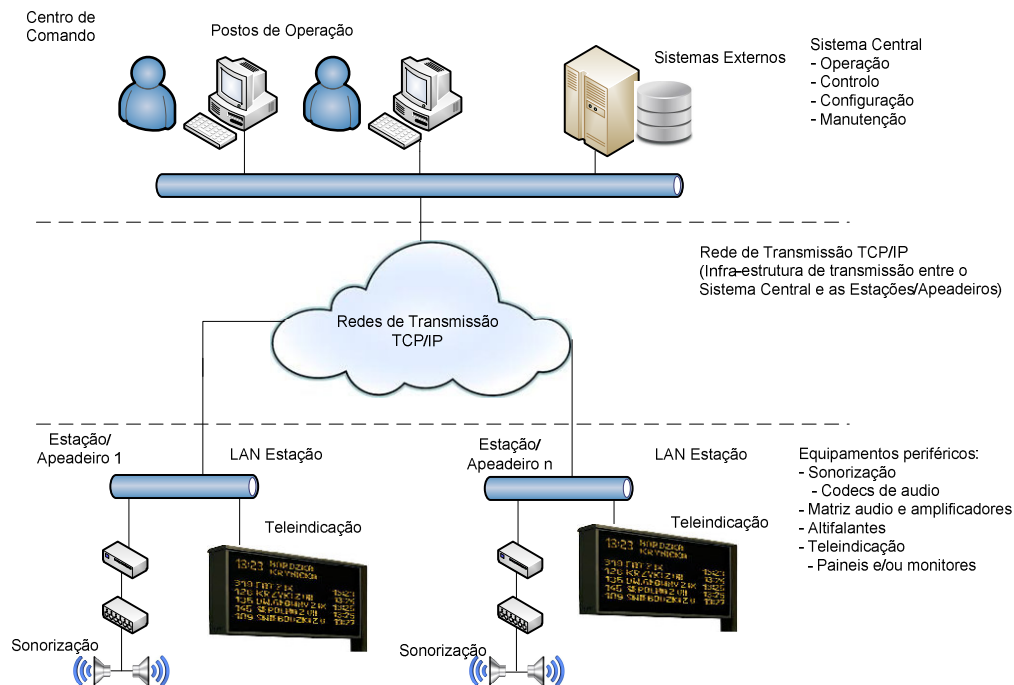


Figura 5 Arquitetura global da solução EFACEC.

O sistema permite disponibilizar aos utentes informação sobre a circulação das composições e informações complementares, introduzidas pelos operadores. A informação de circulação é obtida de um sistema de seguimento dos veículos, por intermédio de um módulo dedicado. A informação de circulação é depois processada automaticamente e distribuída pelos periféricos existentes nas estações [13].

O operador dispõe de uma interface gráfica de fácil utilização que lhe permite efetuar as ações que se seguem sobre equipamentos visuais (painéis ou monitores):

- Enviar mensagens escritas para um painel ou grupo de painéis seleccionados.
- Seleccionar os painéis que afixam informações de circulação.
- Configurar grupos de painéis usados frequentemente.

- Selecionar diferentes *templates* para o mesmo painel, de acordo com a quantidade ou tipo de informação a mostrar (uma linha de texto, uma linha de texto e uma figura, várias linhas de texto, etc.).
- Alinhar e formatar o texto em cada linha.
- Importar *templates*, figuras ou tipos de letra a utilizar nas mensagens a criar.
- Transferir configurações para os painéis, sem interrupção do serviço.

No que toca a equipamentos sonoros, as ações que podem ser efetuadas são:

- Estabelecer chamadas de voz, em tempo real, entre o posto de operação e uma ou mais estações.
- Difundir, por comando dos operadores mensagens previamente gravadas.
- Selecionar as zonas que difundem mensagens automáticas, a partir da informação obtida pela interface de seguimento de composições.
- Gravar mensagens por intermédio do microfone do posto de operação.
- Adicionar mensagens previamente gravadas.
- Transferir mensagens novas para as estações e armazená-las localmente.
- Adicionar músicas ao sistema e transferi-las para as estações.
- Difundir música ambiente nas estações selecionadas.
- Criar *playlists* com as músicas existentes e agendar a sua difusão

O SIP da EFACEC resulta da integração de dois sistemas habitualmente distintos – a tele-indicação (informação visual) e a sonorização (informação sonora). Esta integração resulta numa racionalização dos recursos, uma vez que os dois sistemas partilham a mesma base de informação [13].

Devido às necessidades do mercado, a empresa desenvolveu o EFARail, uma plataforma integrada de supervisão e controle dos vários sistemas necessários à exploração de uma linha metro-ferroviária.

O EFARail (Figura 6) é uma avançada plataforma de operação e gestão para ambientes metro-ferroviários, integrando num mesmo ambiente de operação diferentes subsistemas, de acordo com as necessidades do cliente [14].

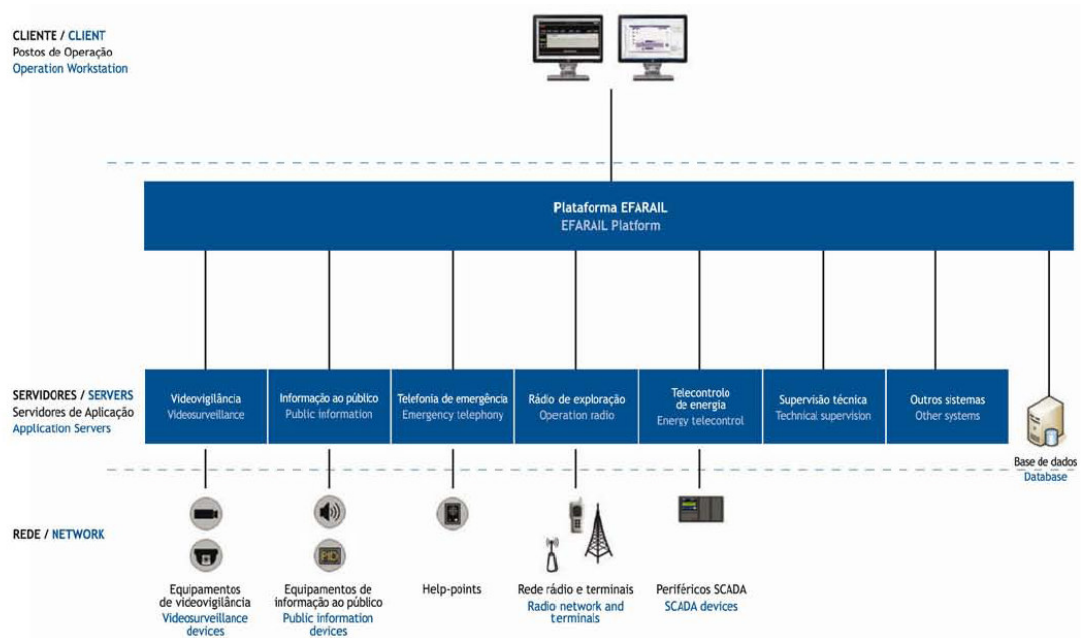


Figura 6 EFARail plataforma de operação e gestão para ambientes metro-ferroviários da EFACEC.

Com uma arquitetura modular, baseada em standards de comunicação, esta plataforma permite a integração num posto de operação unificado, sistemas heterogéneos tais como: controlo de energia, supervisão técnica, videovigilância, controlo de acessos, informação ao público, interfonia, rádio de exploração, assim como a integração com sistemas de terceiros. Deste modo, é conseguido um aumento da eficiência operacional, através de uma visão global da infraestrutura e recursos existentes [14].

3. TECNOLOGIA DE WEB SERVICES PARA A CONSTRUÇÃO DE SISTEMAS DE INFORMAÇÃO AO PÚBLICO

As recomendações internacionais, como as referidas no capítulo 2, sugerem que as soluções a adotar no desenvolvimento de Sistemas de Informação ao Público, devem garantir a interoperabilidade entre módulos de *software* e equipamentos de diferentes fabricantes independentemente das tecnologias que cada um utiliza.

As principais tecnologias que permitem este tipo de interoperabilidade são CORBA, Java RMI, *Web Services*, DCOM, XML-RPC. A principal característica entre elas reside no facto de todas serem independentes da plataforma e da linguagem de desenvolvimento, isto é, permitem a comunicação entre diferentes módulos independentemente da linguagem ou plataforma em que foram desenvolvidos.

A tabela seguinte descreve de uma forma resumida as tecnologias mencionadas [15]:

Tabela 1 Descrição das tecnologias utilizadas em sistemas distribuídos.

TECNOLOGIA	DESCRIÇÃO
Java RMI	O Java RMI (<i>Remote Method Invocation</i>) foi desenvolvido pela divisão Java-Soft da <i>Sun Microsystems</i> , sendo basicamente um mecanismo de RPC (<i>Remote Procedure Calls</i>) orientado a objetos. Isto é, permite a invocação de métodos remotos associados a objetos desenvolvidos na linguagem Java.
CORBA	O CORBA é um padrão criado pelo OMG (<i>Object Management Group</i>) para permitir a interação entre aplicações heterogêneas em ambientes também heterogêneos, isto é, permite a interação entre aplicações desenvolvidas em diversas linguagens de programação que estão a ser executadas em diferentes máquinas (também heterogêneas) conectadas a uma rede de dados.
DCOM	O DCOM (<i>Distributed Component Object Model</i>) é uma arquitetura para objetos distribuídos desenvolvida pela Microsoft. Tem por base o COM (<i>Component Object Model</i>) que define as interfaces de objetos. O DCOM estende o COM com a definição de uma camada de procedimentos remotos permitindo que os objetos possam ser chamados remotamente.
Web Services	Os <i>Web Services</i> são aplicações modulares, auto descritas e acessíveis através da internet. As tecnologias usadas para a construção de <i>Web Services</i> são baseadas em standards e normas abertas, de entre as quais se destacam o XML (<i>eXtensible Markup Language</i>), o SOAP, o WSDL (<i>Web Service Description Language</i>) e o UDDI (<i>Universal Description, Discovery and Integration</i>).

As tecnologias Java RMI, CORBA, DCOM não serão abordadas nesta dissertação. O trabalho final realizado e os resultados obtidos são baseados na tecnologia dos *Web Services*.

Um *Web Service* é uma aplicação identificada por um URI (*Uniform Resource Identifier*), cujas interfaces e ligações são definidas e descritas utilizando XML. A sua descrição pode ser descoberta por outros sistemas. Estes sistemas podem então interagir com o *Web Service* da forma prescrita pela sua descrição, utilizando mensagens em XML transmitidas por protocolos de Internet.

Pode-se classificar um *Web Service* como um tipo de arquitetura que funciona através de protocolos abertos (HTTP e SOAP) e que respondem a pedidos HTTP vindos de qualquer ponto e plataforma conectada na Internet.

A ideia fundamental de um modelo baseado em *Web Services* é que estes devem ser componentes distribuídos, facilmente acessíveis através dos protocolos normalizados como o HTTP.

3.1. ARQUITETURA DE UM WEB SERVICE

Os modelos básicos da arquitetura dos *Web Services* preveem a interação entre três sistemas: o fornecedor do serviço, o registo do serviço e o consumidor do serviço.

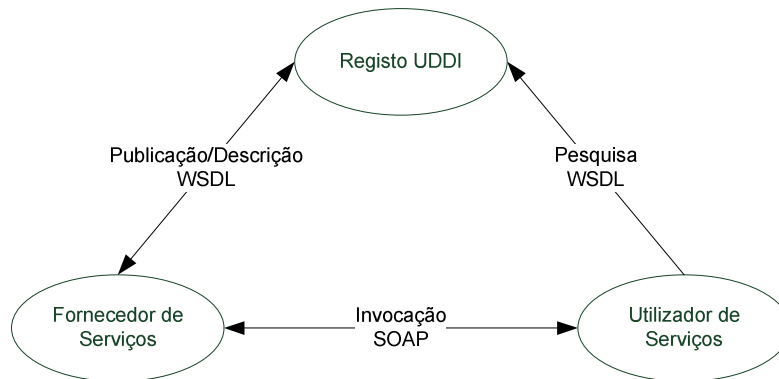


Figura 7 Arquitetura de interação com um *Web Service*.

A Figura 7 ilustra a arquitetura básica de um *Web Service*, sendo que o seu ciclo de vida compreende quatro fases distintas (descrição, publicação, descoberta e invocação) [15]:

- **Descrição** – É o processo pelo qual um *Web Service* descreve a sua interface para que os clientes possam indagar sobre as suas funcionalidades. A interface encontra-se especificada em WSDL e descreve todas as funcionalidades disponibilizadas, assim como os tipos de mensagens que podem ser enviadas e recebidas.
- **Publicação** – É o processo, opcional, através do qual um fornecedor dá a conhecer a existência dos seus *Web Services*, efetuando o registo destes num repositório UDDI.
- **Pesquisa** – É o processo, opcional, através do qual um cliente descobre os *Web Services* pretendidos para invocação pela pesquisa dinâmica de um repositório UDDI.
- **Invocação** – É o processo pelo qual o cliente pede que um serviço seja executado. A invocação envolve uma interação entre o cliente e o serviço através do envio e da receção de mensagens SOAP.

A arquitetura dos *Web Services* está subdividida em vários níveis, devidamente estruturados, para permitirem a execução das operações descritas anteriormente. Nesta arquitetura, as camadas superiores assentam nas funcionalidades disponibilizadas pelas

camadas inferiores. A Figura 8 ilustra a relação entre as várias tecnologias utilizadas pela arquitetura [15].

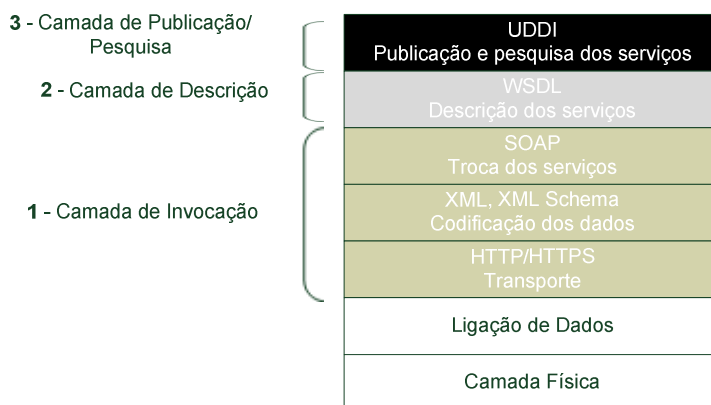


Figura 8 Arquitetura dos *Web Services*.

A camada de invocação é composta pelas subcamadas de transporte, codificação e de troca de mensagens. A subcamada de transporte é aquela sobre a qual todas as outras subcamadas assentam, sendo responsável por transportar as mensagens entre os clientes e os *Web Services*. As mensagens podem ser transportadas por vários protocolos, tais como o HTTP(S), o SMTP (*Simple Mail Transfer Protocol*) e o FTP (*File Transfer Protocol*). Dada a sua simplicidade e divulgação o HTTP tornou-se no protocolo mais usado para o transporte de dados.

A subcamada de codificação de dados tem por base a linguagem XML, uma “metalinguagem” que define um conjunto de regras para a criação de uma linguagem de marcação para codificar os dados de um documento particular ou de uma mensagem. A subcamada de troca de mensagens é suportada pelo protocolo SOAP. Este protocolo define o formato das mensagens utilizadas durante a comunicação entre clientes e serviços.

A segunda camada da arquitetura dos *Web Services* - camada de descrição - fornece uma solução baseada em XML para definir as interfaces dos serviços, os seus dados, os tipos de mensagens que podem ser trocadas, as operações fornecidas, os modelos de interação e os protocolos utilizados [15].

A última camada - camada de publicação/pesquisa - fornece um conjunto de funcionalidades que permite a publicação e a pesquisa de *Web Services*. O componente fundamental desta camada é o UDDI, um mecanismo de registo e descoberta de serviços na *Web*, usado

para armazenar e categorizar informações sobre serviços e para obter referências para interfaces de *Web Services* [15]. A Tabela 2 expõe sumariamente o objetivo de cada uma das camadas analisadas.

Tabela 2 Resumo das normas e protocolos associados aos *Web Services*.

CAMADA	TECNOLOGIA	DESCRIÇÃO
Publicação e Pesquisa	UDDI	Os <i>Web Services</i> são publicados num registo UDDI para que as aplicações clientes possam pesquisar serviços
Descrição dos Serviços	WSDL	Os <i>Web Services</i> usam uma linguagem de descrição chamada WSDL. Uma aplicação cliente necessita de obter o WSDL de um serviço antes de o invocar.
Troca de Mensagens	SOAP	Uma mensagem SOAP é um documento XML que encapsula os dados trocados entre um cliente e um serviço.
Codificação dos Dados	XML	É a linguagem usada para descrever os protocolos associados aos <i>Web Services</i> .
Transporte	HTTP	É o protocolo de comunicação usado para transportar os dados entre clientes e serviços.

3.2. PROTOCOLOS E LINGUAGENS ASSOCIADOS AOS WEB SERVICES

3.2.1. XML

O XML é a base em que os *Web Services* são construídos. Este é uma norma para a troca de dados na *Web*, e é uma linguagem orientada a dados semi-estruturados proposta para solucionar problemas de integração. É também uma linguagem de marcação pois fornece uma forma de adicionar informação sobre os dados. Isso permite uma declaração mais precisa do conteúdo. As principais características são a sua simplicidade, a associação de dados com metadados e a clareza dos documentos criados [15].

3.2.2. XML-RPC

O XML-RPC é um protocolo simples que permite que *software* que esteja a ser executado em diferentes ambientes possa fazer chamadas de procedimentos remotos através da internet. Este protocolo utiliza dois padrões da indústria: XML para a codificação das mensa-

gens e HTTP para o transporte das mesmas. Este protocolo é o antecessor do protocolo SOAP na construção de *Web Services* [16].

3.2.3. SOAP

O SOAP (*Simple Object Access Protocol*) baseia-se numa invocação remota de um método e para tal necessita especificar o endereço do componente, o nome do método e os argumentos para esse método. Estes dados são formatados em XML com determinadas regras e enviados normalmente por HTTP para esse componente. Não define ou impõe qualquer semântica, quer seja o modelo de programação, quer seja a semântica específica de implementação [15].

O SOAP é um protocolo RPC que funciona sobre HTTP, FTP ou SMTP, de forma a ultrapassar as restrições de segurança/firewall normalmente impostas aos sistemas clássicos de RPC (RMI, DCOM, CORBA/IIOP) suportando mensagens XML.

3.2.4. WSDL

O WSDL é um documento escrito em XML que descreve as funcionalidades de um *Web Service*, os tipos de dados e protocolos que podem ser utilizados para enviar e receber mensagens. As operações e as mensagens são descritas de forma abstrata e são associadas a um protocolo de rede e formato de mensagem [15].

O WSDL é extensível para permitir a descrição dos serviços e suas mensagens, independentemente dos formatos de mensagens e dos protocolos de rede que sejam utilizados.

3.2.5. UDDI

O UDDI é um diretório, e especifica um mecanismo centralizado para que os fornecedores de *Web Services* possam publicitar os seus serviços e para que os clientes possam localizar os serviços que necessitem.

O processo de localizar um serviço através de diretórios é denominado de *discovery*. Os diretórios são repositórios que contêm documentos que descrevem serviços. Também possuem funcionalidades tais como a capacidade de pesquisa e o acesso programático por aplicações remotas [15].

3.3. TECNOLOGIAS MAIS UTILIZADAS NA CONSTRUÇÃO DE WEB SERVICES

As tecnologias mais utilizadas no desenvolvimento de *Web Services* são JAX-RPC e JAX-WS (inseridos na plataforma Java EE) e .NET (Microsoft).

3.3.1. JAX-RPC

O JAX-RPC é uma tecnologia criada para a construção de *Web Services* que utilizam RPC e XML. Nesta tecnologia, uma chamada de procedimentos remotos é representada por um protocolo baseado em XML, como o SOAP. O protocolo SOAP define a estrutura do envelope, regras de codificação e convenções para representar chamadas de procedimentos remotos e respectivas respostas. Estas chamadas e respostas são transmitidas como mensagens SOAP (arquivos XML) sobre HTTP [17].

Embora as mensagens SOAP sejam complexas, a API JAX-RPC “esconde” essa complexidade do programador da aplicação.

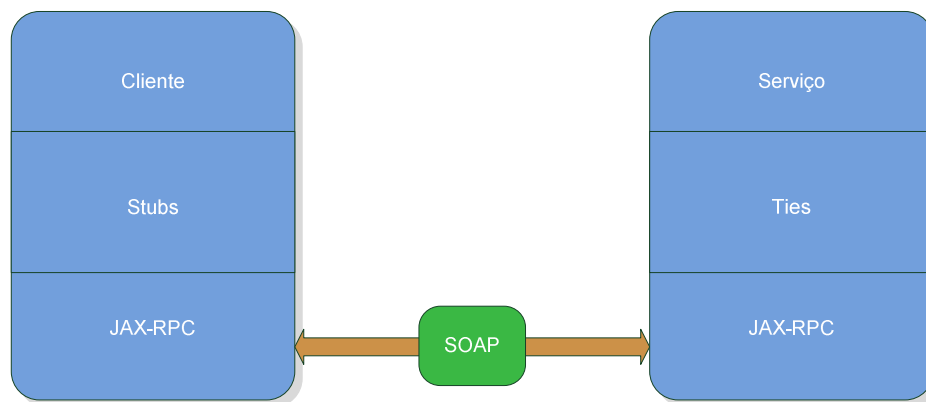


Figura 9 Comunicação Cliente/Servidor utilizando JAX-RPC.

No lado do servidor, o programador especifica os procedimentos remotos através da definição de métodos numa interface escrita na linguagem de programação Java. Para a construção do cliente, é criado um proxy (um objeto local que representa o serviço) e depois simplesmente invoca os métodos através do proxy [17].

Com esta tecnologia, não existe a necessidade de gerar ou analisar mensagens SOAP. É o sistema que converte as chamadas e respostas em mensagens SOAP.

3.3.2. JAX-WS

O JAX-WS permite uma simplificação no desenvolvimento de *Web Services* e clientes, relativamente ao JAX-RPC. Este aborda alguns dos problemas que existem no JAX-RPC, fornecendo suporte a vários protocolos, como SOAP 1.1, SOAP 1.2 e XML [18].

A especificação WSDL 1.1 define uma ligação HTTP, que é um meio pelo qual se pode enviar mensagens XML sobre o protocolo HTTP sem a utilização do protocolo SOAP. No JAX-RPC esta ligação foi ignorada, já no JAX-WS esta funcionalidade é suportada.

O JAX-WS permite efetuar chamadas assíncronas e síncronas ao servidor. As chamadas assíncronas são efetuadas recorrendo a um modelo de *pooling* ou através de *callbacks*.

O modelo de mapeamento de dados utilizado no JAX-WS é o JAXB (*Java Architecture for XML Binding*). O JAXB é uma API que fornece uma forma rápida e conveniente para efetuar a ligação entre esquemas XML e representações Java, tornando mais fácil para os programadores Java a incorporação de dados XML e funções de processamento em aplicações Java [19].

3.3.3. .NET

A plataforma .NET é um produto da Microsoft que permite a criação de aplicações multi-camada. Os principais componentes para a construção de *Web Services* na plataforma .NET são: uma aplicação cliente, um *Web Service*, vários ficheiros com a lógica do serviço, um servidor *Web* para correr a aplicação e opcionalmente um servidor de base de dados para o armazenamento e acesso aos dados da aplicação [20].

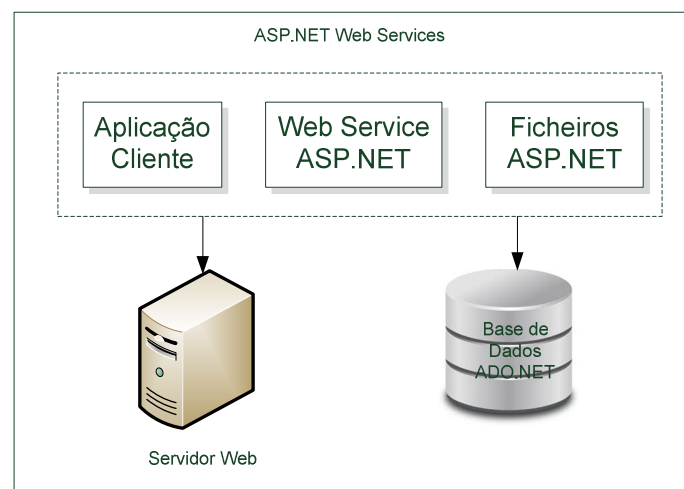


Figura 10 Arquitetura de um Web Service construído na plataforma .NET.

Os *Web Services* escritos para esta plataforma podem ser desenvolvidos em várias linguagens, como C#, Visual Basic .NET e ASP.NET [21].

4. IMPLEMENTAÇÃO

Com o objetivo de testar a plataforma de gestão EFARail da EFACEC, foi proposta a criação de um simulador que permitisse a validação da comunicação da plataforma de gestão com um SIP genérico.

4.1. REQUISITOS DO SISTEMA

Os requisitos do sistema englobam fatores como a finalidade do sistema, tipo de utilizadores e ambiente de execução. Sendo assim, a arquitetura a ser adotada precisa atender às seguintes características:

- Modularidade: o sistema deve ser desenvolvido em camadas, havendo uma interface de comunicação bem definida entre as mesmas.
- Manutenção: o sistema deve adotar padrões de documentação e codificação bem definidos.
- Extensibilidade: as ferramentas utilizadas no processo de desenvolvimento do sistema deverão ser de acesso aberto e, preferencialmente, gratuito. Da mesma forma, bibliotecas de *software* de terceiros que forem utilizadas deverão ser de código aberto.

- Reutilização: a arquitetura do sistema deve ser tal que permita a utilização de classes e componentes noutros projetos, favorecendo o tempo de produção e a qualidade do produto gerado.
- Portabilidade: a fim de garantir portabilidade e independência da plataforma, o sistema deve ser desenvolvido na plataforma Java.

4.2. PROPOSTA DE ARQUITETURA

O simulador desenvolvido, ilustrado na Figura 11, consiste num conjunto de módulos distintos que comunicam entre si segundo o modelo cliente-servidor.

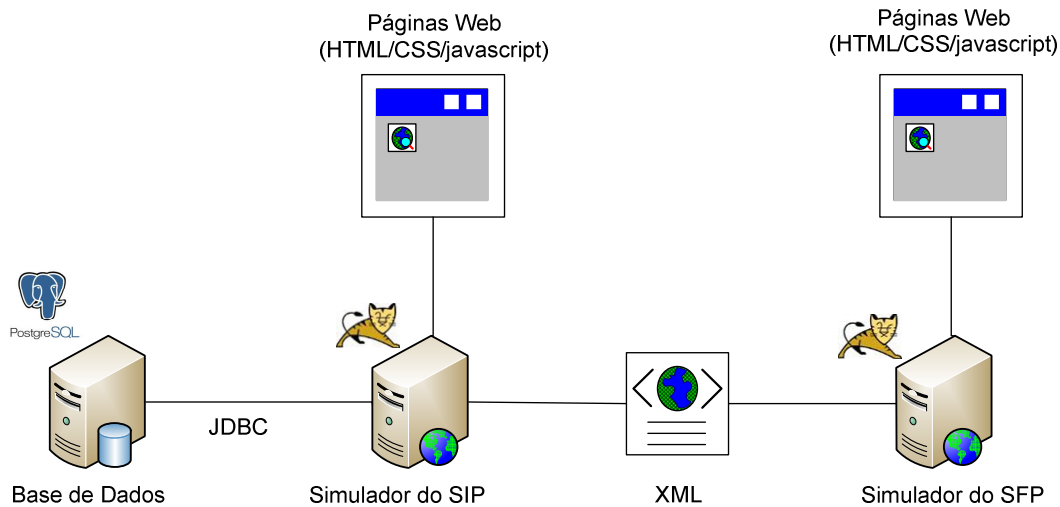


Figura 11 Arquitetura do simulador do SIP.

O **Simulador do SIP** é responsável pela simulação dos equipamentos de informação ao público numa linha metro-ferroviária, o acesso à base de dados, pela disponibilização de uma interface gráfica de manutenção do simulador e do *Web Service* que permite a comunicação do SFP com este.

O **Simulador do SFP** foi construído com o objetivo de se efetuar testes ao Simulador do SIP através do *Web Service*, na ausência da plataforma de gestão da EFACEC.

A **Base de Dados**, armazena toda a informação relativa à constituição da linha metro-ferroviária em simulação (como por exemplo: estação, constituição da estação em termos de zonas e equipamentos associados a cada uma).

Para a elaboração destes três módulos foram utilizadas as seguintes tecnologias e linguagens de programação:

- Servidor *Apache Tomcat* (versão 7.x);
- Servidor de base de dados PostgreSQL (versão 9.x);
- *Integrated Development Environment* (IDE), Netbeans (versão 7.2);
- Tecnologia *JavaServer Pages* (JSP);
- Linguagens de programação Java, *Javascript*, HTML (*HyperText Markup Language*) e CSS (*Cascading Style Sheet*);

4.3. ARQUITETURA DO SOFTWARE

4.3.1. BASE DE DADOS

A base de dados de suporte ao simulador está representada no Anexo A, sendo constituída por trinta tabelas. Esta foi desenvolvida no sistema de gestão de base de dados PostgreSQL que, para além de ser *software* de código fonte aberto, proporciona muitas funcionalidades modernas como: comandos complexos, chaves estrangeiras, integridade transacional entre outros.

As principais tabelas da base de dados implementada são: *t_station*, *t_stationarea*, *t_panel*, *t_soundarea*, *t_log*, *t_visualmessage*, *t_soundmessage*. Na Tabela 3 é feita uma descrição de cada uma destas tabelas.

Tabela 3 Descrição das principais tabelas da base de dados implementada.

TABELA	DESCRIÇÃO
<i>t_station</i>	Armazena informação referente a uma estação como o indentificador e o nome da estação.
<i>t_stationarea</i>	Armazena informação referente a uma zona de estação.
<i>t_panel</i>	Armazena informação de equipamento visual como: nome do equipamento, identificador do equipamento, identificador da zona de estação a que pertence, estado de funcionamento e o estado das afixações automáticas.
<i>t_soundarea</i>	Armazena informação de equipamento sonoro como: nome do

TABELA	DESCRIÇÃO
	equipamento, identificador do equipamento, identificador da zona de estação a que pertence e estado de funcionamento.
t_log	Armazena informação das operações efetuadas sobre o Simulador do SIP.
t_visualmessage	Armazena informação das mensagens destinadas a equipamento visual.
t_soundmessage	Armazena informação das mensagens destinadas a equipamento sonoro.

4.3.2. ARQUITETURA DE SOFTWARE DO SIMULADOR DO SIP

O Simulador do SIP foi desenvolvido utilizando o servidor de aplicações *Web* Apache Tomcat, este é um servidor *open source* da Apache Software Foundation que implementa as tecnologias *Java Servlet* e *Java Server Pages*.

Foi escolhido como *container* do Simulador não só por se tratar de uma ferramenta gratuita como também por ser uma aplicação muito popular e com provas dadas de fiabilidade e segurança.

Como se pode ver pela Figura 12, o Simulador do SIP é constituído por quatro blocos de *software* que permitem a interação com os vários elementos do sistema.



Figura 12 Diagrama de blocos do Simulador do SIP.

O bloco denominado por `Web Service`, é constituído pela classe `SimSMM.java`, que contém um conjunto de métodos que permitem a comunicação da plataforma de gestão da EFACEC com o Simulador do SIP.

A `Aplicação Web` permite a disponibilização de uma interface gráfica, com a qual o operador interage através de um *browser* recolhendo informações e efetuando operações sobre os vários equipamentos do SIP.

Esta aplicação é constituída por um conjunto de ficheiros JSP que estão divididos em três diretórios:

- Equipamento Sonoro;
- Equipamento Visual;
- SIMSIP.

O diretório `Equipamento Sonoro` contém os ficheiros que permitem obter informações e efetuar operações sobre uma determinada zona de som. Na Tabela 4 apresenta-se uma breve descrição dos ficheiros contidos neste diretório.

Tabela 4 Descrição dos ficheiros contidos no diretório Equipamento Sonoro.

FICHEIRO	DESCRIÇÃO
EquipSonoro.jsp	Apresenta toda a informação relativa a uma zona de som, como por exemplo: mensagens que estão a ser difundidas, estado de funcionamento da zona de som, estado de funcionamento da música ambiente.
ChangeEquipState.jsp	Permite a alteração do estado de funcionamento de uma determinada zona de som.
ConfigMusicState.jsp	Permite a alteração do estado da música ambiente de uma determinada zona de som.
GetSoundLog.jsp	Obtém da base de dados os registos das operações efetuadas sobre uma determinada zona de som.
ReleaseSoundArea.jsp	Elimina toda a informação de uma determinada zona de som contida em memória.

O diretório `Equipamento Visual` contém os ficheiros que permitem obter informações e efetuar operações sobre um determinado equipamento visual. Na Tabela 5 faz-se uma breve descrição dos ficheiros contidos neste diretório.

Tabela 5 Descrição dos ficheiros contidos no diretório Equipamento Visual.

FICHEIRO	DESCRIÇÃO
EquipVisual.jsp	Apresenta toda a informação relativa a um equipamento visual, como por exemplo: mensagens que estão a ser afixadas, estado de funcionamento do equipamento visual, estado das afixações automáticas.
ChangeEquipState.jsp	Permite a alteração do estado de funcionamento de um determinado equipamento visual.
ConfigAutoModeConnection.jsp	Permite a alteração do estado das afixações automáticas de um determinado equipamento visual.
GetVisualLog.jsp	Obtém da base de dados os registos das operações efetuadas sobre um determinado equipamento visual.
ReleasePanel.jsp	Elimina toda a informação contida em memória de um determinado equipamento visual.

O diretório `SIMSIP` contém os ficheiros que constroem a página de entrada da aplicação *Web*, permitindo uma visualização em árvore da estrutura da linha que está a ser simulada

e ainda um menu com tópicos que permitem obter informação que está armazenada na base de dados do simulador. Na Tabela 6 faz-se a descrição de cada um desses ficheiros.

Tabela 6 Descrição dos ficheiros contidos no diretório SIMSIP.

FICHEIRO	DESCRIÇÃO
MenuPrincipal.jsp	Apresenta a página de entrada com um menu que disponibiliza um conjunto de funcionalidades que permitem obter informações armazenadas na base de dados.
Tree.jsp	Cria uma árvore com a estrutura da linha metro-ferroviária que está a ser simulada. Através desta árvore é possível selecionar um determinado equipamento visual ou sonoro.
GetAllPlayLists.jsp	Obtém as playlists armazenadas na base de dados.
GetHolidays.jsp	Obtém as datas referentes a feriados gravados na base de dados.
GetLogs.jsp	Obtém todos os registos gravados na base de dados relativos ao funcionamento do simulador.
GetSoundTempEvent.jsp	Obtém os eventos temporários sonoros gravados na base de dados.
GetVisualTempEvent.jsp	Obtém os eventos temporários visuais gravados na base de dados.
PreRecordedSoundMsg.jsp	Obtém as mensagens sonoras gravadas na base de dados.
PreRecordedVisualMsg.jsp	Obtém as mensagens visuais gravadas na base de dados.

O bloco de *software* denominado por *Lógica da Aplicação* contém um conjunto de classes que permitem a simulação do Sistema de Informação ao Público; na Tabela 7 é feita uma descrição de cada uma das classes.

Tabela 7 Classes que constituem o bloco Lógica da Aplicação.

CLASSE	DESCRIÇÃO
ANSWER.class	Estrutura que define a resposta obtida de cada equipamento.
EQ_STATE.class	Estrutura que possui a informação do estado de cada equipamento.
MESSAGE.class	Estrutura que define uma mensagem de afixação.
TIME.class	Estrutura que define uma data/hora.
EQUIP.class	Estrutura que define um equipamento visual ou sonoro.
STATION.class	Estrutura que define uma estação.

CLASSE	DESCRIÇÃO
<code>STATIONAREA.class</code>	Estrutura que define uma zona de estação.
<code>TEMP_MSG.class</code>	Estrutura que define um evento temporário visual ou sonoro.
<code>SoundArea.class</code>	Estrutura com os campos e métodos necessários para a definição de um equipamento sonoro.
<code>Panel.class</code>	Estrutura com os campos e métodos necessários para a definição de um equipamento visual.
<code>OperationLog.class</code>	Esta classe contém os métodos que permitem efetuar o registo das operações realizadas sobre o simulador.
<code>SimulationOfVisualEquipment.class</code>	Classe com os vários métodos necessários ao controle de um equipamento visual.
<code>SimulationOfSoundEquipment.class</code>	Classe com os vários métodos necessários ao controle de um equipamento sonoro.
<code>PRE_RECORDED_MSG.class</code>	Estrutura que define a informação associada a uma mensagem pré-gravada, esta é utilizada tanto para mensagens visuais como sonoras.
<code>PLAYLIST.class</code>	Estrutura que contém a informação de configuração de uma playlist.
<code>LOG_DATA.class</code>	Estrutura que contém a data e a operação efetuada sobre o simulador.
<code>ACTIVE_PLAYLIST.class</code>	Estrutura que agrega informação da playlist atualmente ativa e da playlist que se encontra a tocar.

As classes principais deste bloco são `SimulationOfVisualEquipment.class` e `SimulationOfSoundEquipment.class`. A primeira é constituída por um conjunto de métodos que simulam as várias operações efetuadas sobre equipamentos visuais, como por exemplo: afixação de mensagens (quer sejam de tempo real ou pré-gravadas), remoção de mensagens de um painel, ligar/desligar a afixação de informação de circulação, entre outras. A segunda é constituída por um conjunto de métodos que simulam as operações efetuadas sobre equipamentos sonoros, que passam pela difusão de mensagens (de tempo-real ou pré-gravadas), cancelamento da difusão de mensagens, difusão de música ambiente, entre outras.

As restantes classes são estruturas de dados que permitem o armazenamento em memória de informação relativa aos equipamentos visuais e sonoros que constituem o SIP. Esta informação é constituída por vários elementos como por exemplo identificadores de equi-

pamentos, estado de funcionamento dos equipamentos, nome das estações, identificadores das estações, identificadores de mensagens, etc.

O último bloco de *software* denominado Acesso à Base de Dados, é constituído pelas classes que estão descritas na Tabela 8, estas permitem a acesso à base de dados para a consulta, inserção e eliminação de informação relativa a uma determinada linha metro-ferroviária.

Tabela 8 Classes que constituem o bloco Acesso à Base de Dados.

CLASSE	DESCRIÇÃO
<code>HandleDataBase.class</code>	Classe com os métodos e atributos que permitem a inserção, remoção e consulta da informação contida na base de dados.
<code>ConnectDataBase.class</code>	Classe com métodos que permitem a ligação à base de dados.

4.3.3. ARQUITETURA DE SOFTWARE DO SIMULADOR DO SFP

Para a construção do Simulador do SFP foi utilizado o mesmo servidor de aplicações *Web* que foi utilizado para o Simulador do SIP, pelas razões descritas anteriormente. O Simulador do SFP foi desenvolvido criando um *proxy* (um objeto local que representa o *Web Service* disponibilizado pelo Simulador do SIP) e um conjunto de ficheiros JSP. O conjunto de ficheiros JSP está dividido em três diretórios (Figura 13):

- Equipamento Sonoro
- Equipamento Visual
- SIMSFP

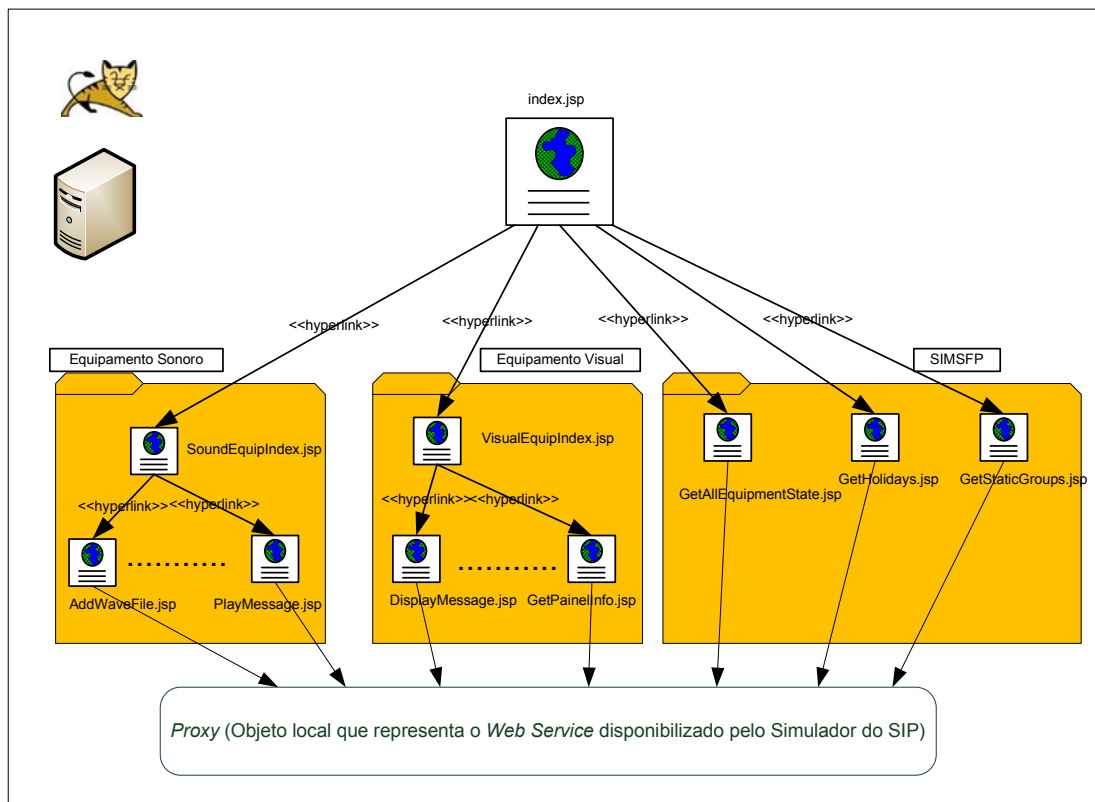


Figura 13 Arquitetura do software do Simulador do SFP

No diretório `Equipamento Sonoro` estão os ficheiros JSP que permitem o acesso aos métodos disponibilizados pelo *Web Service* relativos aos equipamentos sonoros. No diretório `Equipamento Visual` estão os ficheiros JSP que permitem o acesso aos métodos disponibilizados pelo *Web Service* mas neste caso relativos aos equipamentos visuais.

Por último o diretório `SIMSFP` contém os ficheiros JSP que permitem aceder aos métodos destinados à obtenção de informação como os feriados, o estado dos equipamentos e os grupos de equipamentos gravados no Simulador do SIP.

4.4. INTERFACE COM O UTILIZADOR

4.4.1. INTERFACE GRÁFICA DO SIMULADOR DO SIP

Para se poder visualizar toda a informação referente a uma linha metro-ferroviária que se pretenda simular, foi necessário construir uma interface gráfica.

Na Figura 14 é apresentada a página de entrada da interface gráfica do Simulador do SIP e o tipo de informação disponibilizada.



Figura 14 Página de entrada da interface gráfica do simulador do SIP.

Esta é constituída por duas áreas distintas, uma assinalada a verde (1) e a outra assinalada a azul (2).

Na área (1) é disponibilizada uma árvore com a estrutura das várias estações de uma determinada linha metro-ferroviária. Selecionando uma estação é possível visualizar as várias zonas de estação e dentro de cada zona de estação listar os equipamentos sonoros e visuais existentes. De forma a simplificar a apresentação da informação é possível seleccionar filtros que permitem optar por visualizar apenas equipamentos sonoros ou equipamentos visuais.

A área (2) é constituída por uma *toolbar* na parte superior, sendo a parte inferior reservada para a disponibilização de diversa informação referente ao sistema e aos equipamentos sonoros ou visuais. A *toolbar* é constituída por várias opções que permitem obter informações do simulador que estão armazenadas na base de dados. Essas opções são:

- LOGs – Permite obter todos os registos referentes a operações efetuadas sobre o simulador. Como se pode ver na Figura 15, esta opção disponibiliza um formulário no qual pode ser inserido um intervalo de tempo. Ao submeter estes dados o simulador imprime uma tabela com a informação da data, hora e tipo de operação efetuada.

REDE GERAL

EQUIPAMENTO VISUAL

EQUIPAMENTO SONORO

INTRODUZIR UM INTERVALO DE TEMPO PARA OBTER O REGISTO DE OPERAÇÕES

Data de Inicio: 2013-08-10 00:00 Data de Fim: 2013-08-20 23:59 SUBMETTER

REGISTO DE OPERAÇÕES EFETUADAS

DATA	HORA	OPERAÇÃO
2013-08-19	12:16:01.0	Desligar Afixação de Informação de Circulação
2013-08-19	12:15:48.0	Ligar Afixação de Informação de Circulação
2013-08-14	17:22:25.0	Alterar estado de funcionamento para Reservado
2013-08-14	17:22:20.0	Desligar Música Ambiente
2013-08-14	17:12:09.0	Desligar Afixação de Informação de Circulação
2013-08-14	17:12:00.0	Ligar Afixação de Informação de Circulação
2013-08-14	17:11:56.0	Libertar Painel
2013-08-14	17:11:53.0	Ligar Afixação de Informação de Circulação
2013-08-14	17:11:49.0	Alterar estado de funcionamento para Livre
2013-08-14	17:11:44.0	Alterar estado de funcionamento para Aviso
2013-08-14	17:11:39.0	Alterar estado de funcionamento para Alarme
2013-08-14	17:11:36.0	Alterar estado de funcionamento para Reservado
2013-08-14	17:11:31.0	Ligar Afixação de Informação de Circulação

Figura 15 Informação obtida através da opção LOGs.

- FERIADOS – Disponibiliza uma lista com a data de todos os feriados inseridos no sistema. Na Figura 16 pode ver-se o resultado da informação obtida ao seleccionar esta opção.

REDE GERAL

EQUIPAMENTO VISUAL

EQUIPAMENTO SONORO

FERIADOS CONFIGURADOS NO SIMULADOR

ANO	MES	DIA
2013	6	25
2013	7	26
2013	7	29
2013	9	2
2013	9	25
2013	10	25
2013	11	15

Figura 16 Informação obtida através da opção FERIADOS.

- PLAYLISTs – Disponibiliza uma lista com o nome e o identificador das playlists disponíveis no sistema para serem difundidas nos equipamentos sonoros - Figura 17.

ID	Nome
1	Playlist 1
2	Playlist 2
3	Playlist 3
4	Playlist 4
5	Playlist 5
6	Playlist 6
7	Playlist 7
8	Playlist 8
9	Playlist 9
10	Playlist 10

Figura 17 Informação obtida através da opção **PLAYLISTs**.

- **MENSAGENS PRÉ-GRAVADAS** – Disponibiliza uma lista com as mensagens gravadas no sistema. Esta opção possui um submenu para se seleccionar qual o tipo de mensagens (visuais ou sonoras) - Figura 18.

ID da Mensagem	Texto	Ref. xpto
1	Teste 1	url1
2	Teste 2	url2
3	Teste 3	url3
4	Teste 4	url4
5	Teste 5	

Figura 18 Informação obtida através da opção **MENSAGENS PRÉ-GRAVADAS**.

- **EVENTOS TEMPORÁRIOS** – Disponibiliza uma lista com os eventos temporários gravados no sistema. Esta opção possui um submenu para se seleccionar qual o tipo de eventos (visuais ou sonoros) - Figura 19.

EVENTOS TEMPORÁRIOS VISUAIS GRAVADOS NO SISTEMA

ID	DATA DE INICIO	DATA DE FIM	DIAS	CONTINUIDADE
1	0002-11-30 00:00:00.0	0002-11-30 00:00:00.0		NÃO
2	0002-11-30 00:00:00.0	0002-11-30 00:00:00.0		NÃO

Figura 19 Informação obtida através da opção **EVENTOS TEMPORÁRIOS**.

- **GRUPOS DE EQUIPAMENTOS** – Disponibiliza uma lista com os grupos de equipamentos gravados no sistema. Esta opção possui um submenu para se selecionar quais os grupos de equipamentos (visuais ou sonoros) - Figura 20.

GRUPOS DE EQUIPAMENTOS SONOROS

ID	NOME DO GRUPO	TIPO DE GRUPO	EQUIPAMENTOS
2	Grupo 2	Sonoro	6, 9, 11
4	Grupo 4	Sonoro	1, 6, 7, 9

Figura 20 Informação obtida através da opção **GRUPOS DE EQUIPAMENTOS**.

Para além da informação disponibilizada através da *toolbar*, ainda existe a possibilidade de se obter informação relativa a cada equipamento. Como se pode ver na Figura 21 quando se seleciona um equipamento da árvore é apresentada informação sobre esse, incluindo o estado de funcionamento do equipamento, o estado das afixações automáticas, as mensagens que estejam a ser afixadas, etc.



Figura 21 Informação obtida quando se seleciona um equipamento visual.

4.4.2. INTERFACE GRÁFICA DO SIMULADOR DO SFP

Para se testar os vários métodos disponibilizados pelo *Web Service* foi criada uma interface gráfica. Nesta, como se pode verificar na Figura 22, existe um menu do lado esquerdo com as seguintes opções: Equipamento Visual, Equipamento Sonoro, Estado dos Equipamentos, Feriados, Grupo de Equipamentos.

Cada uma destas opções permite que se aceda aos vários métodos disponibilizados pelo *Web Service* para que se possam enviar instruções ao Simulador do SIP.

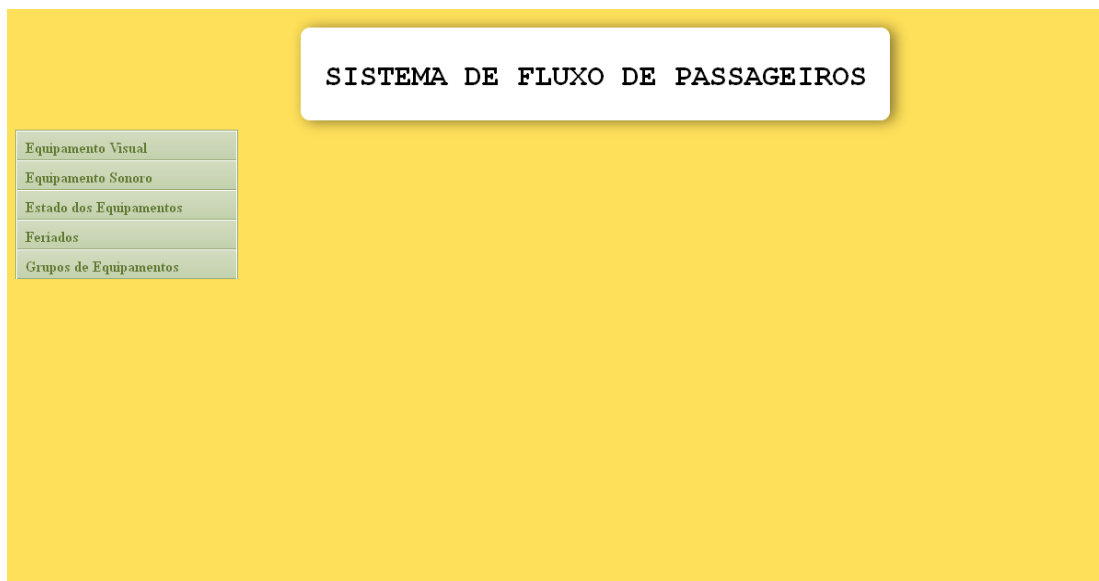


Figura 22 Página de entrada da interface gráfica do Simulador do SFP.

Selecionando a opção *Equipamento Visual*, é apresentada uma página (Figura 23) com um menu do lado esquerdo que disponibiliza um conjunto de opções que permitem que se

enviem várias instruções aos equipamentos visuais do Simulador do SIP, como por exemplo *Afixar Mensagem*. Como se pode ver na Figura 23, é apresentado um formulário correspondente à opção *Afixar Mensagem*, o qual é constituído por um conjunto de campos, necessários para o envio desta instrução ao Simulador do SIP.

EQUIPAMENTO VISUAL

AFIXAR MENSAGEM

ID DO PAINEL

2
4
5
6

INTRODUZA NO ID -1 PARA MENSAGEM DE TEMPO REAL

ID DA MENSAGEM

ID: -1	TEXT	Mensagem Teste 1	REF	90956078
ID: -1	TEXT	Mensagem Teste 2	REF	394509345
ID: -1	TEXT	Mensagem Teste 3	REF	0084056
ID: -1	TEXT	Mensagem Teste 4	REF	980968456

FORMATAÇÃO DA MENSAGEM 0-SEM FORMATAÇÃO/1-RODAR/2-PISCAR

FORMATAÇÃO: 1

ALINHAMENTO DA MENSAGEM 0-ESQUERDA/1-CENTRO/2-DIREITA

ALINHAMENTO: 1

ID DO POSTO DE OPERAÇÃO: 12

AFIXAR MENSAGEM

Figura 23 Formulário disponibilizado pela opção *Afixar Mensagem*.

Preenchido o formulário e clicando sobre o botão “AFIXAR MENSAGEM”, os dados são enviados através do *Web Service* para o simulador do SIP o qual se encarrega de tratar a respetiva informação recebida.

A opção *Equipamento Sonoro* disponibiliza uma página (Figura 24) com idêntica à da Figura 23 mas neste caso com opções que permitem que se enviem instruções destinadas aos equipamentos sonoros. Como exemplo temos o formulário correspondente à opção *Difundir Mensagem Pré-Gravada*.

EQUIPAMENTO SONORO

Difundir Mensagem TempReal
Terminar Difusão TempReal
Difundir Mensagem PreGravada
Parar Difusão Msg Pré-Gravada
Difundir Playlist
Inserir Evento Temporário
Obter Eventos Temporários
Cancelar Evento
Ativ/Desativ Estado Event Temp
Obter Estado Event Temp
On/Off Estado Musica Amb
Obter Estado Musica Ambiente
Adicionar Ficheiro Audio
Apagar Ficheiro de Audio
Obter Ficheiro de Audio

DIFUNDIR MENSAGEM PRE-GRAVADA

ID DAS ZONAS DE SOM

2

3

5

7

ID DAS MENSAGENS

1

2

3

4

NUMERO DE REPETIÇÕES

2

ID DO OPERADOR

13

DIFUNDIR MENSAGEM

Figura 24 Formulário disponibilizado pela opção Difundir Mensagem Pré-Gravada.

As opções Feriados, Grupos de Equipamentos e Estado dos Equipamentos permitem que se obtenha informação que está armazenada no Simulador do SIP. Como exemplo temos na Figura 25 a informação obtida quando se seleciona a opção Estado dos Equipamentos.

ESTADO DOS EQUIPAMENTOS

Equipamento Visual
Equipamento Sonoro
Estado dos Equipamentos
Feriados
Grupos de Equipamentos

ID	TIPO	ESTADO
1	Visual	Livre
2	Visual	Reservado
3	Visual	Livre
4	Visual	Alarme
5	Visual	Reservado
6	Visual	Aviso
7	Visual	Livre
8	Visual	Livre
9	Visual	Reservado
10	Visual	Alarme
11	Visual	Livre
12	Visual	Aviso
13	Visual	Livre
14	Visual	Livre

Figura 25 Informação obtida quando se seleciona a opção Estado dos Equipamentos.

5. TESTES FUNCIONAIS

Para a realização dos testes foi utilizado o *software* soapUI [22], uma ferramenta *open source* escrita em Java cuja principal função é consumir e testar *Web Services*. O soapUI fornece um conjunto robusto de recursos para testar *Web Services*, não só durante o desenvolvimento, mas também para testar a validade das implementações.

A utilização desta aplicação obriga à criação de um projeto (menu “File → New soapUI-Project”), onde é indicado o diretório ou URL no qual se encontra o ficheiro WSDL com a descrição do *Web Service* (Figura 26).

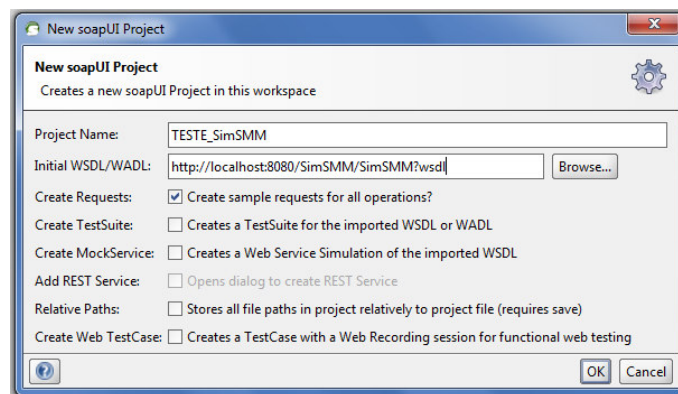


Figura 26 Criação de um novo projeto.

Introduzidos os dados, o programa importa o ficheiro WSDL e apresenta, uma árvore com todos os métodos disponíveis no *Web Service* (Figura 27).

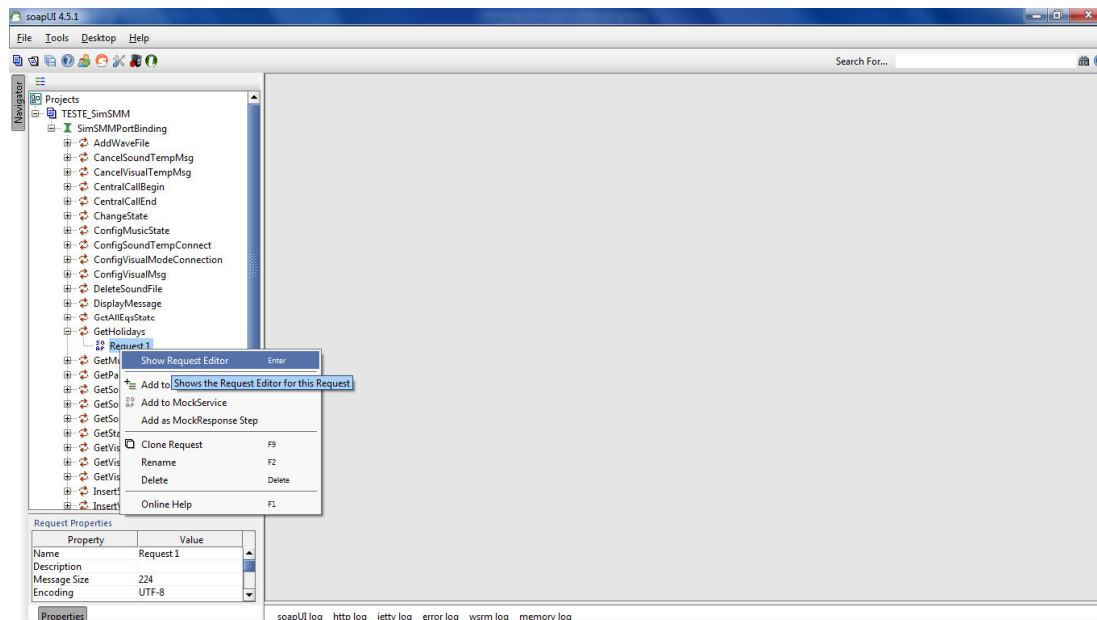


Figura 27 Árvore com os métodos disponibilizados pelo *Web Service*.

Através desta árvore é possível seleccionar um qualquer método para se efetuar testes. Para exemplificar seleccionou-se o método `GetHolidays` o qual permite obter todos os feriados gravados no simulador do SIP.

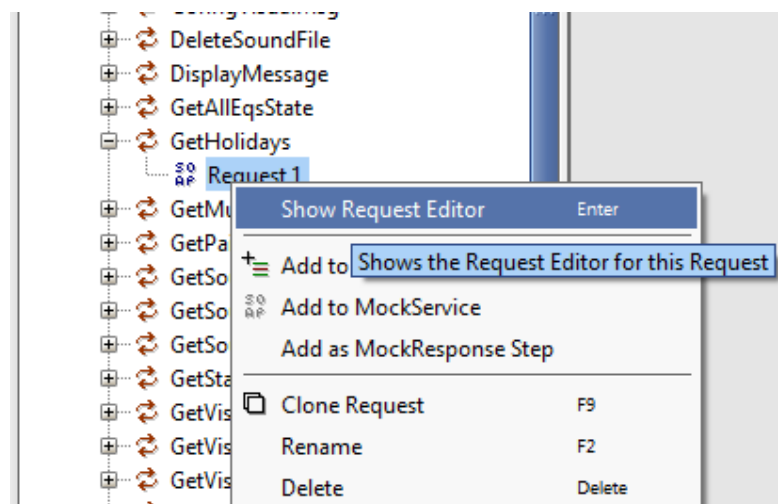


Figura 28 Seleção do método `GetHolidays`.

Como se pode verificar pela Figura 28, ao seleccionar o método `GetHolidays` aparece uma extensão denominada "*Request 1*". Ao clicar com o botão direito do rato sobre esta

extensão aparece uma caixa de diálogo, selecionando-se a primeira opção “*Show Request Editor*” aparece uma janela dividida em dois (Figura 29), do lado esquerdo temos a mensagem SOAP que foi enviada ao Simulador do SIP e do lado direito a mensagem SOAP com a resposta do simulador.

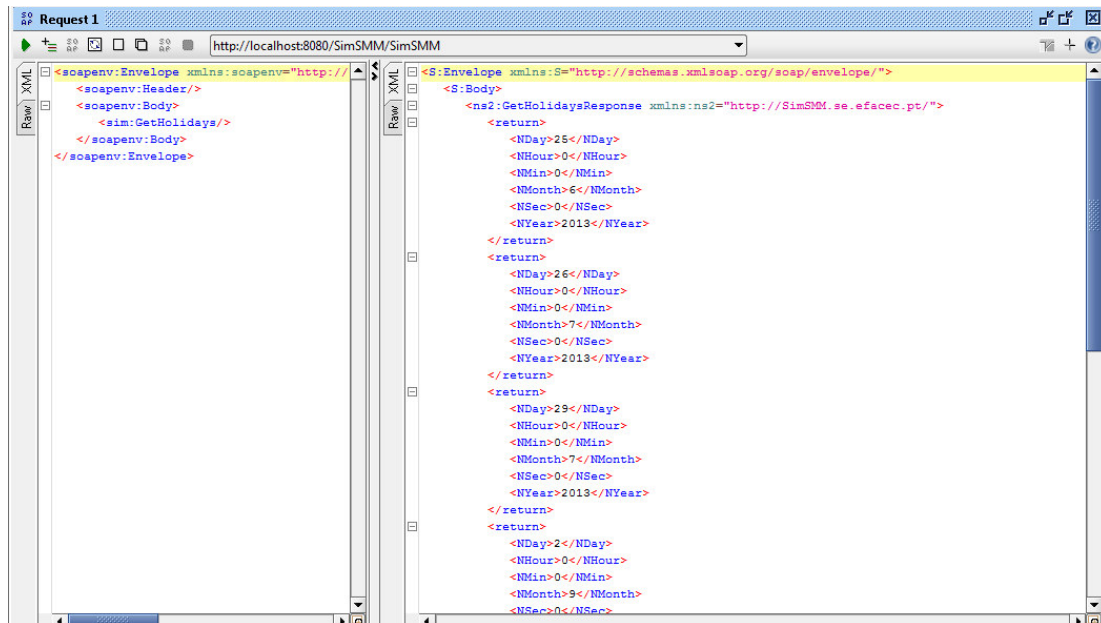


Figura 29 Mensagens SOAP trocadas.

Foram feitos testes a todos os métodos disponíveis na árvore utilizando esta ferramenta. Esta ao contrário do Simulador do SFP, permite verificar o conteúdo das mensagens SOAP trocadas entre o Simulador do SIP e o cliente do *Web Service*, que tanto pode ser o Simulador do SFP ou o próprio SFP da plataforma de gestão EFARail. No Anexo B é apresentado um teste ao método `DisplayMessage` para uma melhor compreensão do tipo de informação que se obtém utilizando a ferramenta soapUI e o Simulador do SFP. Comparando a Figura 31 com a Figura 32 é notória a abstração que se obtém com o Simulador do SFP, tornando mais fácil o preenchimento dos campos necessários para o envio do comando `DisplayMessage` ao Simulador do SIP. Em contrapartida, perde-se informação que é relevante para o desenvolvimento do *Web Service*, tanto na deteção de eventuais erros de funcionamento como na melhoria do mesmo.

6. CONCLUSÕES

No âmbito desta dissertação desenvolveram-se ferramentas que têm como objetivo facilitar o teste em ambiente laboratorial de módulos a ser integrados numa plataforma de apoio à gestão de redes metro-ferroviárias, nomeadamente os seus componentes associados aos sistemas de informação ao público. A solução desenvolvida numa plataforma *Web* acrescenta flexibilidade e mobilidade ao sistema eliminando as instalações necessárias de uma aplicação de *desktop* e fornecendo acesso às funcionalidades do sistema através de um *browser* instalado em qualquer computador.

A utilização de *Web Services* no desenvolvimento da aplicação proposta confere, para além das vantagens referidas no Capítulo 3, as características necessárias em termos de interoperabilidade e comunicação entre o Sistema de Fluxo de Passageiros da plataforma EFARail e um Sistema de Informação ao Público da EFACEC ou de terceiros. Outro ponto a salientar na utilização de *Web Services* está nas ferramentas utilizadas para a sua construção, estas mascaram a complexidade proporcionando maior rapidez de execução e uma solução eficiente e satisfatória.

Um aspeto importante na construção de simuladores está no facto de estes permitirem que se efetue testes em laboratório que ajudam na deteção de anomalias de funcionamento e de

implementação. Desta forma reduz-se os riscos do sistema apresentar anomalias depois de instalado no projeto para o qual foi construído.

A ferramenta desenvolvida foi testada do ponto de vista das funcionalidades fornecidas e comunicação entre módulos de *software*, tendo sido atingidos os objetivos propostos.

Como perspectivas de evolução futura, reconhece-se a necessidade de melhoria das interfaces gráficas dos simuladores no sentido de as tornar mais intuitivas para o operador. A possibilidade de efetuar simulações de várias linhas em simultâneo, é também uma funcionalidade que deverá ser acrescentada ao Simulador do SIP em versões futuras.

Referências Documentais

- [1] IMTT, “Sistemas de informação ao público”, Coleção de Brochuras Técnicas/Temáticas de apoio à elaboração de Planos de Mobilidade e Transportes, Março de 2011, Disponível em: <http://www.conferenciamobilidade.imtt.pt/tema14.php>.
- [2] INFOPOLIS 2, “Needs of Travellers an Analysis Based on the Study of their Tasks and Activities”, 1998 a 2000, Disponível em: ftp://ftp.cordis.europa.eu/pub/telematics/docs/tap_transport/infopolis2_d3.pdf.
- [3] WYG International, “Pre-Investment study of the passenger information system for the airport Poznan-Lawica SP. Z O.O.”, 2011, Disponível em: http://www.champions-project.de/public_docs/4.2.6%20Pre-Investment%20study%20Poznan_web.pdf.
- [4] SUNDGREN, Bo, “What is a public information system”, Dept. of Information Technology and Media Mid, Sweden University, 2005, Disponível em: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.127.6022>.
- [5] THE JAVA TOTURIALS, “Introduction to JAXB”, 2013, Disponível em: <http://docs.oracle.com/javase/tutorial/jaxb/intro/..>
- [6] RSL, “Interactive Terminals”, 2013, Disponível em: <http://www.rslweb.co.uk/innerpage.php?mid=1&parent=1&id=138>.
- [7] TECNAU, “Dynamic Public Information (PIMS)”, 2013, Disponível em: http://www.tecnau.it/trasporti_en.htm.
- [8] INFOPOLIS 2, “Review of Current Passenger Information Systems”, 1998 a 2000, Disponível em: ftp://ftp.cordis.europa.eu/pub/telematics/docs/tap_transport/infopolis2_d1.pdf.
- [9] IMTT, “Manual de Tecnologias de Informação e Comunicação”, 2010, Disponível em: http://www.imtt.pt/sites/IMTT/Portugues/TransportesRodoviaros/Documents/Manuais%20Forma%C3%A7%C3%A3o%20Inicial%20Motoristas/Manual_Tecnologias_Informacao_Comunicacao_FIC.pdf.
- [10] SIRI, “Service Interface for Real Time Information”, 17 de Abril 2011, Disponível em: <http://user47094.vs.easily.co.uk/siri/overview.htm>.
- [11] SIRI, “Management Overview – White Paper”, 2005, Disponível em: <http://user47094.vs.easily.co.uk/siri/documentation.htm>.
- [12] INTEGRAIL, “Publishable Final Activity Report”, IGR-P-DAP-156-07, 1 de Janeiro de 2005, Disponível em: <http://www.integrail.info/>.

- [13] OLIVEIRA, Paula, “Sistema de Informação ao Público para Ambientes Metro-Ferrovios” FER XXI, Edição Nº 35, maio de 2007.
- [14] EFARAIL, “Gestão Integrada de Sistemas Metro-Ferrovios”, Disponível em: http://www.efacec.pt/presentationlayer/efacec_produto_01.aspx?idioma=1&idProduto=312.
- [15] CARDOSO, Jorge, “Programação de Sistemas Distribuídos em Java”, FCA, 2008. ISBN 978-972-722-601-6.
- [16] LAURENT, Simon St., JOHNSTON, Joe, DUMBILL, Edd, “Programming Web Applications with XML-RPC”, O’Reilly, First Edition June 2001, ISBN: 0-596-00119-3.
- [17] THE J2EE 1.4 TUTORIAL, “Building Web Services with JAX-RPC”, Disponível em: <http://docs.oracle.com/javaee/1.4/tutorial/doc/JAXRPC.html>.
- [18] TECNAU, “Dynamic Public Information (PIMS)”, 2013, Disponível em: http://www.tecnau.it/trasporti_en.htm.
- [19] THE JAVA TUTORIALS, “Introduction to JAXB”, 2013, Disponível em: <http://docs.oracle.com/javase/tutorial/jaxb/intro/>.
- [20] “ASP.NET Web Services Architecture”, 2008, Disponível em: http://docs.embarcadero.com/products/rad_studio/radstudio2007/RS2007_helpupdates/HUUpdate4/EN/html/devnet/webservicesov_xml.html#4153502E4E45542057656220536572766963657320417263686974656374757265.
- [21] KACHRU, Sandeep; GEHRINGER, Edward F., “A Comparison of J2EE and .NET as Platforms for Teaching Web Services”, 23 de Outubro de 2004, ISBN 0-7803-8552-7, Disponível em: <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?reload=true&arnumber=1408754&contentType=Conference+Publications>.
- [22] SoapUI, Disponível em: <http://www.soapui.org/>.
- [23] TIDWELL, Doug; SNELL, James; KULCHENKO, Pavel, “Programming Web Services with SOAP”. O’Reilly, 2001. ISBN 0-596-00095-2.
- [24] “Introdução aos Web Services”, Disponível em: https://netbeans.org/kb/docs/websvc/jax-ws_pt_BR.html.

Anexo A. Base de Dados Desenvolvida

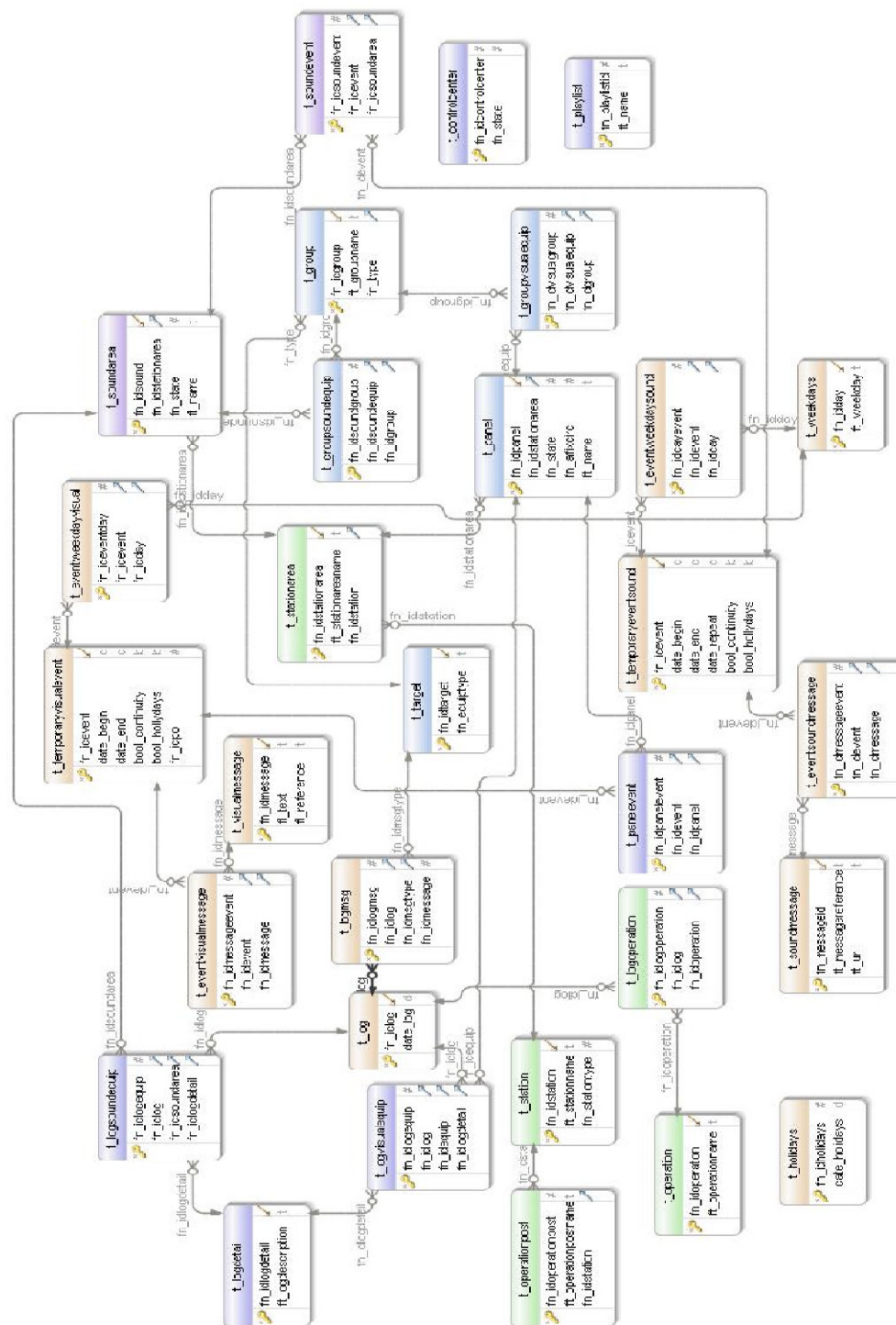


Figura 30 Base de Dados desenvolvida.

Anexo B. Testes efetuados ao método DisplayMessage.

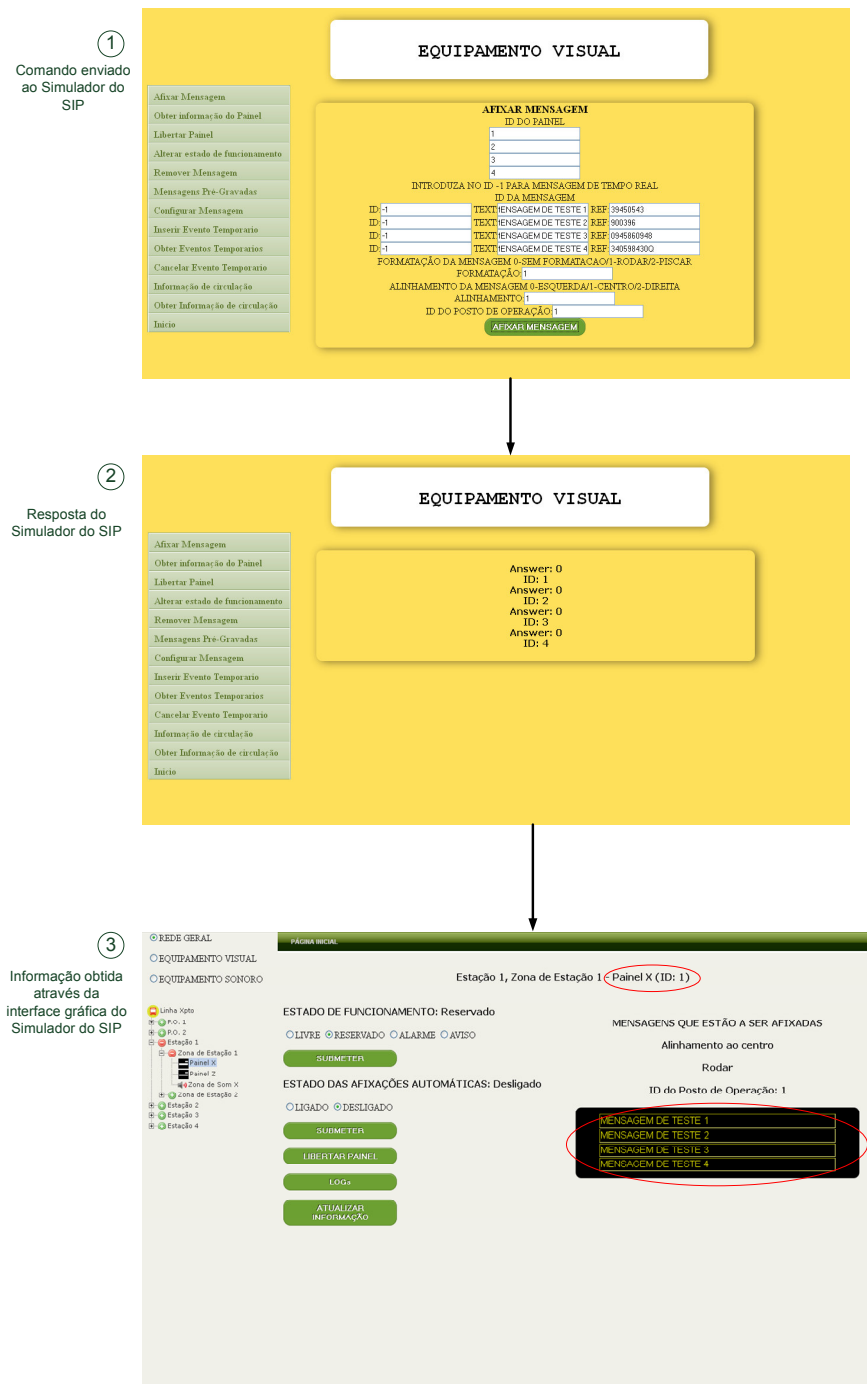


Figura 31 Comando enviado através do Simulador do SFP.

